Facultad de Informática
Universidad Politécnica de Madrid

Master Thesis

Master in Artificial Intelligence Research

Morphological Study of Dendritic Spines

Author:
Biomedical Eng.
Miguel Federico Baguear

Supervisors:
PhD. Luis Baumela
PhD. Pedro Larrañaga

July 2011

Contents

Acknowledgments

- To Luis Baumela, Concepción Bielza and Pedro Larrañaga for their trust an help during all the project.
- To Isabel Fernaud-Espinosa, Ruth Benavides-Piccione and Javier De Felipe from the Cortical Circuits Laboratory for their help, suggestions, biological feedback and for providing us with the Dendritic Spine data-base.
- To Pablo Márquez-Neila for his comments and suggestions.
- Cajal Blue Brain Project Scholarship - BB08002009.
- To my family in Buenos Aires.
- To Farah for supporting and loving me during the distance.

Resumen

La reconstrucción de la estructura de la neurona y la caracterización de las espinas dendríticas es hoy en día una de las áreas de trabajo de mayor interés en la investigación en neurobiología. Las dendritas son estructuras celulares en forma de ramas cuyo objetivo es conducir el impulso nervioso de naturaleza electroquímica procedente de otras neuronas al cuerpo o al soma de la neurona. Las espinas dendríticas son pequeñas protrusiones que aparecen en la superficie (membrana) de la dendrita sobre las cuales se asienta la densidad postsináptica, que es la región donde se concentran los neurotransmisores. Las espinas tienen diferentes formas que pueden variar en el tiempo y cuya dinámica se cree que está íntimamente relacionada con los procesos del aprendizaje y la memoria.

El objetivo de esta tesis de máster fué desarrollar algoritmos que analicen bloques de imágenes de microscopía confocal que ya han sido procesadas manualmente, con la finalidad de realizar una reconstrucción precisa de la estructura tridimensional de las espinas y llevar a cabo un estudio sobre su morfología.

La tesis está organizada de la siguiente forma; en la introducción se mencionarán las bases fisiológicas neuronales de las espinas dendríticas y su clasificacion clásica, luego se explicará brevemente el funcionamiento del microscopio confocal y la extracción manual de las espinas que ha sido realizada por los investigadores del Laboratorio de Circuitos Corticales del Consejo Superior de Investigaciones Científicas (CSIC). En la sección de metodología se explicará el procedimiento manual de extracción de espinas, el algoritmo que hemos implementado para su reconstrucción y corrección; luego se describiran los descriptores de forma y los métodos de clustering que hemos utilizado. En el capítulo 3 se muestran los resultados obtenidos y en el capítulo 4 se discuten los mismos y se plantean posibles mejoras y líneas de investigación posteriores.

Abstract

Reconstruction of the neuron structure and the characterisation of dendritic spines is nowadays a hot topic in neurobiology research. Dendrites are cellular structures whose main objective is to conduct the electrochemical stimulation received from other neural cells to the cell body of the neuron from which the dendrites project. A dendritic spine is a small membranous protrusion from a neuron's dendrite that typically receives input from a single synapse of an axon. Spines have different shapes that can change over time and are believed to be closely related to learning and memory.

The main objective of this thesis is to develop an algorithm to analyse the stack of confocal microscopy images that have been previously manually processed, in order to obtain a precise three dimensional reconstruction of the dendritic spines and study their morphology.

The outline of this thesis is as follows. In the next chapter we introduce the neuronal physiological basis of the dendritic spines and their classical classification. We also briefly explain how the confocal microscope works. In the methodology chapter we expose the manual procedure to extract the dendritic spines done by the researchers of the Cortical Circuits Laboratory, the spine reconstruction algorithm that we have implemented, the 3D shape descriptors and the clustering methods that we have used. In the third chapter we show the obtained results and we end the thesis by discussing them and proposing future research lines.

Chapter 1

Introduction

In Purves (2008) a neuron is defined as an electrically excitable cell that processes and transmits information by electrical and chemical signaling. The basic cellular organization of neurons resembles that of other cells; however, they are clearly distinguished by specialization for intercellular communication. This attribute is apparent in their overall morphology, in the specific organization of their membrane components for electrical signaling, and in the structural and functional intricacies of the synaptic contacts between neurons. The most obvious sign of neuronal specialization for communication via electrical signaling is the extensive branching of neurons. The most salient aspect of this branching for typical nerve cells is the elaborate arborization of dendrites that arise from the neuronal cell body. Dendrites are the primary target for synaptic input from other neurons whereas the axon is a long projection of the neuron that conducts electrical impulses away from the neuron's soma. Dendritic spines are the projections of a neuron that act to conduct the electrochemical stimulation received from other neural cells to the cell body of the neuron from which the dendrites project. A dendritic spine is a small membranous protrusion from a neuron's dendrite that typically receives input from a single synapse of an axon. Spines have different shapes that can change over time and are believed to be closely related with learning and memory (Yuste and Bonhoeffer (2001) and Arellano et al. (2007)). Figure 1.1 shows the structure of a impregnated neuron.

This work is divided into two, the first part focuses on the three dimensional dendritic spine reconstruction from a stack of confocal microscopy images; based on these reconstructions, the second step focuses on the study of the morphology of spines.

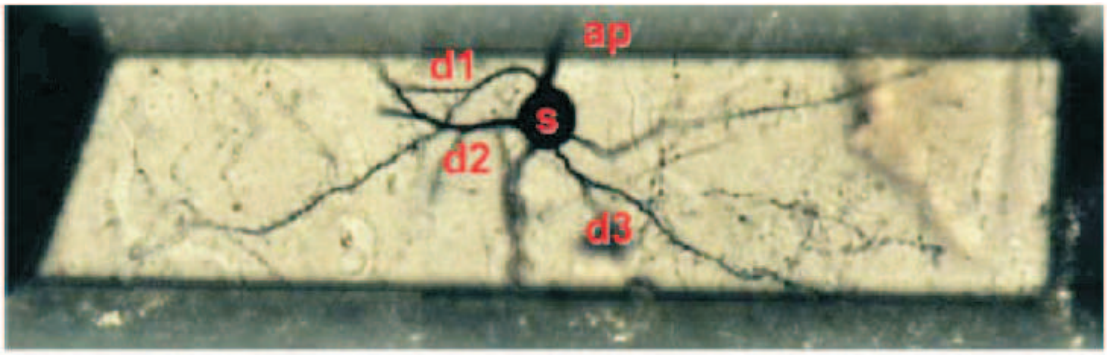

Figure 1.1: Obtained from Arellano et al. (2007). Trimmed resin block containing a selected Golgi-impregnated gold-toned neuron. The soma (s), some basal dendrites (d1-d3) bearing dendritic spines and the beginning of the apical dendrite (ap) are indicated. Scale bar is 24 μm.

1.1 Dendritic Spines

Dendritic spines are responsible for essential functions of the cerebral cortex, as they establish synaptic connections and are the recipient of most excitatory inputs to pyramidal cells, the most abundant cortical neuronal type (Ramón y Cajal (1899)). Yuste and Bonhoeffer (2001) highlights the potential relation between spine shape and synaptic function, a possibility that has become more likely since morphological rearrangements of spines have been found in vitro and in vivo and appears to be associated in some cases with developmental or behavioral plasticity (Arellano et al. (2007)). For example, it is hypothesized that age-related and disease-related declines in cognitive ability are accompanied by decreases in spine density (Fiala et al. (2002)). Figure 1.2 shows a three dimensional rendering of an apical dendritic segment where three dendritic spines, called S1, S2 and S3, are marked.

1.1.1 Dendritic Spines Classification

It is possible to find many different dendritic spines described in the literature. Also, this diversity has been hypothesized as a possible factor influencing spine stability and function. The classical distinction between spines was proposed by Peters and Kaiserman-Abramof (1970), based on optical microscopy and observation of spines in single sections at the electron microscope. They distinguished between stubby, thin and mushroom spines. Importantly, they also indicated that

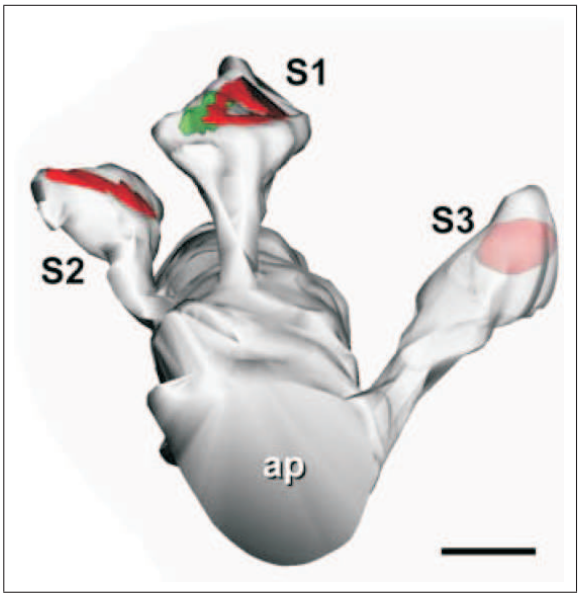

Figure 1.2: Obtained from Arellano et al. (2007). Three dimensional reconstruction of an apical dendritic segment; the rendering has been shifted down to show the synaptic junctions and S3 is partially transparent to show the location of the post synaptic density (PSD). Scale bar is 0.6 μm.

some spines had intermediate forms and were difficult to classify in those types. Arellano et al. (2007) address this topic using a computation three-dimensional reconstruction of the spines. After analyzing the electron microscopy images, they conclude that the morphology of the spines showed a continuum of their variability in shape and size. In other words, this means that no clear sub-grouping of spines could be detected in the distributions of morphological variables (see Figure 1.3).

The present work aims to accurately characterise the morphology of the spines from confocal microscopy images. Starting from about eight thousand manually segmented spines, the first step will be to correct and complete each spine using a computer vision algorithm; the second step aims to cluster the morphological parameters of the spines in order to test the classification proposed by Peters and Kaiserman-Abramof (1970).

1.2 Confocal Laser Scanning Microscopy

The following information is based on the Confocal Laser Scanning Microscopy (CLSM) brochure written by Wilhelm (2010). In the CLSM the complete genera-

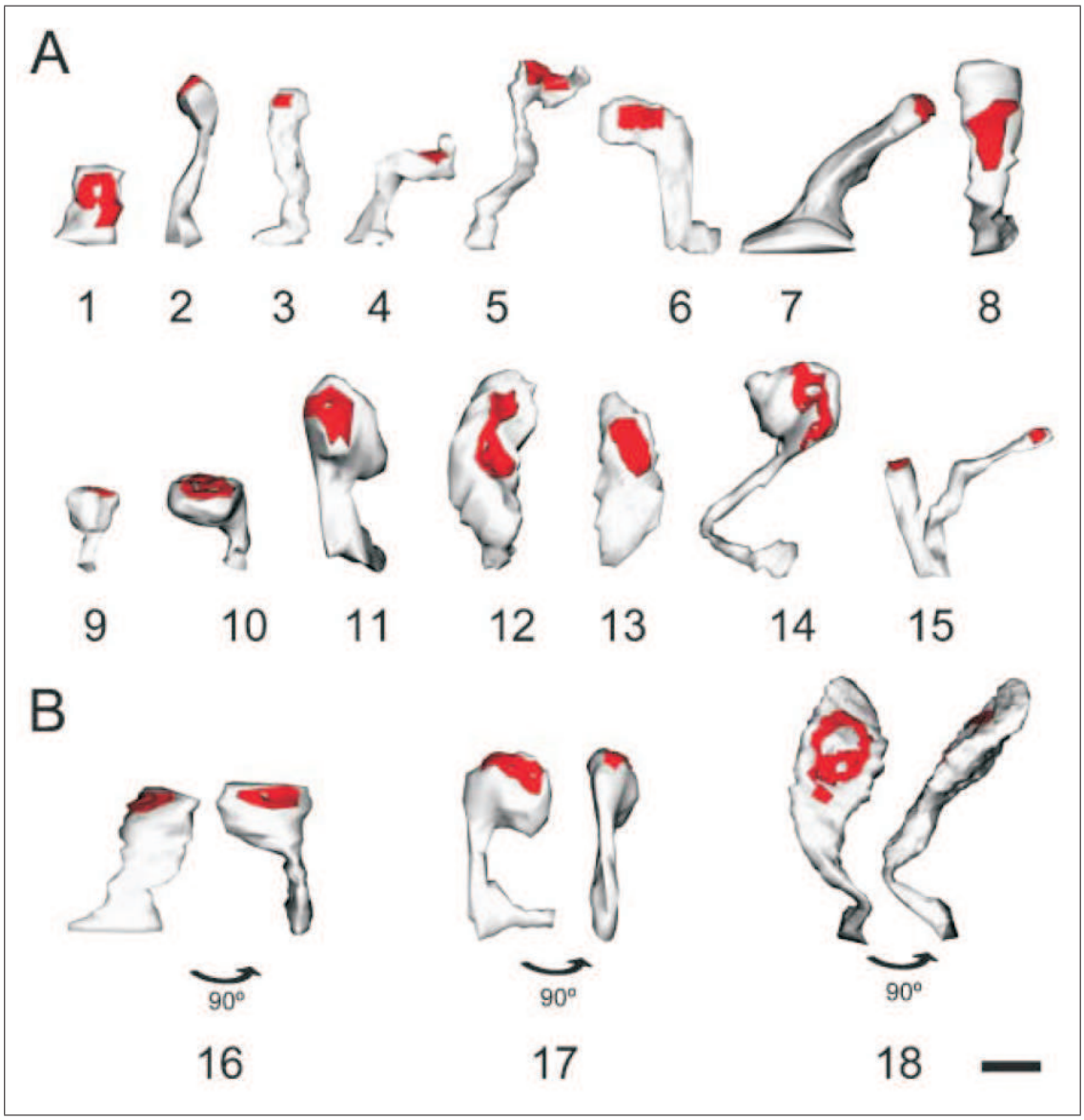

Figure 1.3: Obtained from Arellano et al. (2007). Morphological variability of spines. Three dimensional reconstruction of spines showing the variability in their morphology. (A) Spines showing different types, stubby (1), thin (2), mushroom (9-11), and ramified (15). The author highlight that most reconstructed spines were atypical or intermediate types (3-8, 12-14). (B) Spines appear different depending on the angle of observation. 16-18 illustrate three spines from two points of view after 90 degrees rotation. Scale bar is 0.5 μ m.

tion of the two-dimensional object information from the object plane (focal plane) of a confocal LSM essentially comprises three process steps:

1. Line-by-line scanning of the specimen with a focused laser beam deflected in the X and Y directions by means of two galvanometric scanners.
2. Detection of the fluorescence emitted by the scanned specimen details, by means of a photomultiplier tube (PMT).
3. Digitisation of the object information contained in the electrical signal provided by the PMT (for presentation, the image data are displayed, pixel by pixel, from a digital matrix memory to a monitor screen).

In a conventional light microscope, object-to-image transformation takes place simultaneously and parallel for all object points. By contrast, the specimen in a confocal LSM is irradiated in a pointwise fashion, i.e. serially, and the physical interaction between the laser light and the specimen detail irradiated (e.g. fluorescence) is measured point by point. To obtain information about the entire specimen, it is necessary to guide the laser beam across the specimen, or to move the specimen relative to the laser beam, a process known as scanning.

1.2.1 Optical Slices

With a confocal LSM and its variable pinhole it is therefore possible to exclusively image a thin optical slice out of a thick specimen (typically, up to 100 nm), a method known as optical sectioning. Figure 1.4 shows a slice obtained from a dendrite from a basal neuron.

1.2.2 3rd Dimension

In addition to the possibility to observe a single plane (or slice) of a thick specimen in good contrast, optical sectioning allows a great number of slices to be cut and recorded at different Z-planes of the specimen, with the specimen being moved along the optical axis by controlled increments. The result is a 3D data set, which provides information about the spatial structure of the object. The quality and accuracy of this information depend on the thickness of the slice and on the spacing between successive slices (optimum scanning rate in Z direction is equal to the 0.5x of the slice thickness). By computation, various aspects of the object

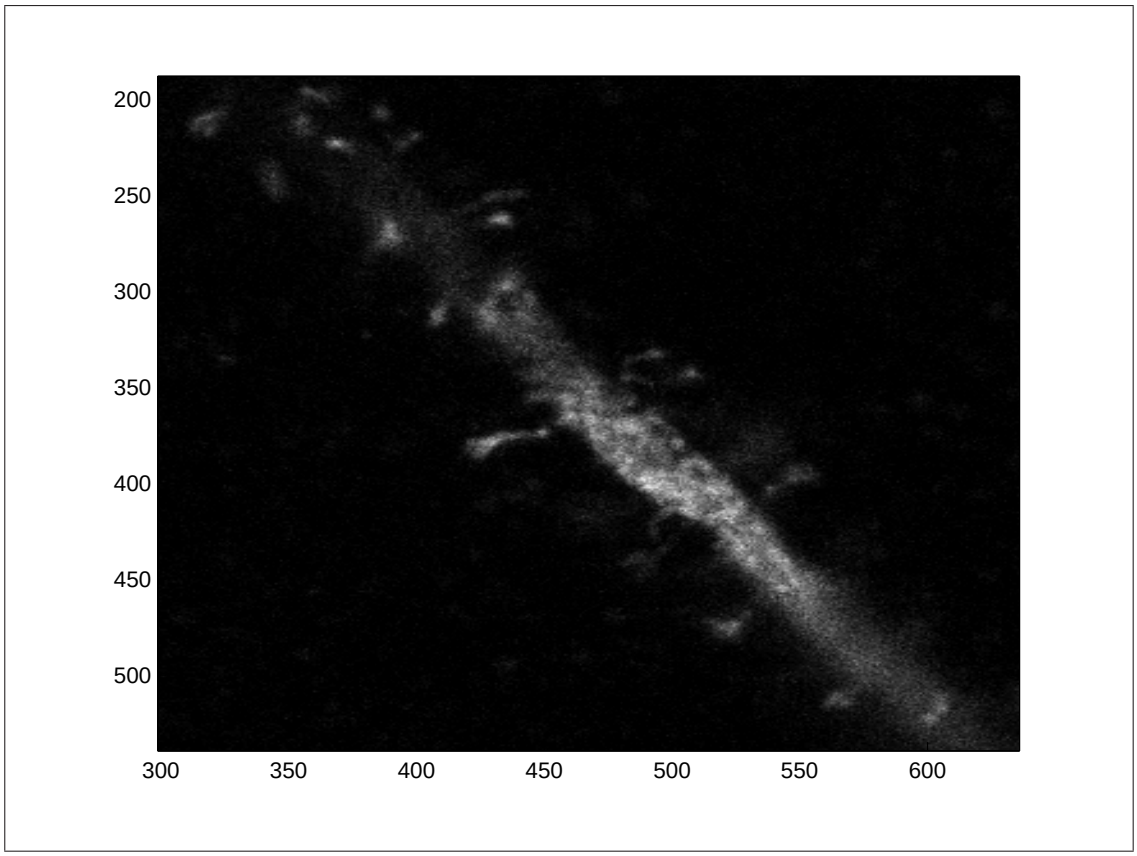

Figure 1.4: Confocal Microscope Image obtained from a basal dendrite segment. Original Image size= 76.8 μ m x 76.8 μ m. Pixel size = 75 nm x 75 nm.

can be generated from the 3D data set (3D reconstruction, sections of any spatial orientation, stereo pairs etc.). Figure 1.2 shows a 3D reconstruction computed from a 3D data set.

Chapter 2

Methodology

2.1 Introduction

As mentioned above, this work aims to accurately characterise the morphology of dendritic spines obtained from confocal microscopy images. It is important to highlight that the original data has been obtained by a semiautomatic procedure that took almost 3 years of work. Starting from eight thousand manually segmented spines, the first step was to reconstruct and complete each spine using a computer vision algorithm; the second step aimed to cluster the morphological parameters of the spines in order to improve the classification proposed by Peters and Kaiserman-Abramof (1970).

2.2 Tissue Samples

Tissue samples were obtained from two healthy human subjects. Subject 1 (M16) was forty years old and subject 2 (IF6) was 85 years old. Both samples have been obtained from the same cortical area and neuronal cortex. From these samples basal and apical types of dendrits have been chosen. In the first ones, the origin of the dendrits is in the top of the soma. In the later, the origin of the dendrits is in the bottom of the soma (Figure 1.1). Fixed human brain tissue sections were used to perform the injections. Neurons were individually injected with Lucifer Yellow (LY). To enhance the signal, immunohistochemical methods were used, with a secondary antibody coupled to Alexa 488. The fluorescence of this molecule was captured with the aid of a Confocal Laser Microscope.

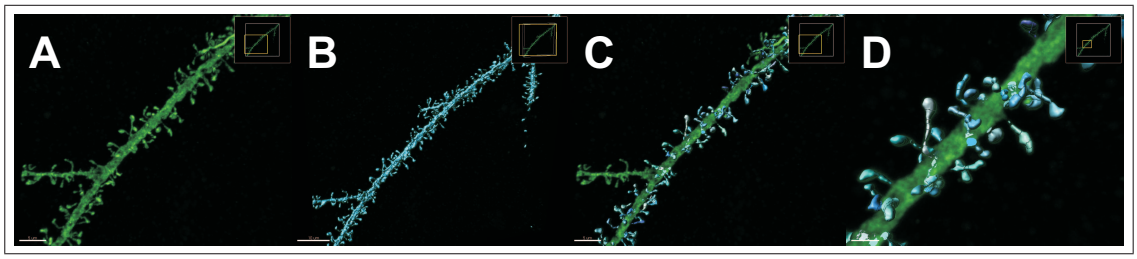

Figure 2.1: Image of an apical dendrite shown with the Imaris Software®. A) Dendrite body. B) Isosurface of the dendrite. C) Semi-automatically segmented dendritic spines. D) Closer view of the semi-automatically segmented dendritic spines. Original image size= 76.8 μm x 76.8 μm. Pixel size = 75 nm x 75 nm

Manual Extraction of Dendritic Spines

The confocal laser microscopy stack of images was processed using a specialise image processing software (Imaris ®- Bitplane AG). After this process that includes image oversampling, filtering and segmentation, a 3D volume of the whole specimen model is obtained (Figure 2.1). It is important to notice that due the irregular properties of the biological samples, the intensity of the fluorescence measured by the CLSM is not constant across all the tissue and thin or small parts that emit few photons can not be distinguished from noise. In order to work with the dendritic spines a manual process is needed to correctly select and numerate the objects of interest. This manual procedure involves selecting the image intensity threshold for each spine or pieces of spine, marking the point of insertion in the dendritic body and marking other points used to measure the length of the dendritic spine. As the result of this semiautomatic task a Virtual Reality Modelling Language (VRML) and an Excel (Microsoft Office) files are generated. The VRML includes the mesh information of each dendritic spine while the Excel file includes information obtained by the Imaris software such as volume, area, length or centre of mass (just to mention a few).

Figure 2.2 shows a correctly segmented dendritic spine by the software, while Figure 2.3 shows a broken dendritic spine formed by two separated pieces. In order to fully characterise the whole set of spines with the same features we have developed a Spine Reconstruction Algorithm (SRA) to fix the broken spines.

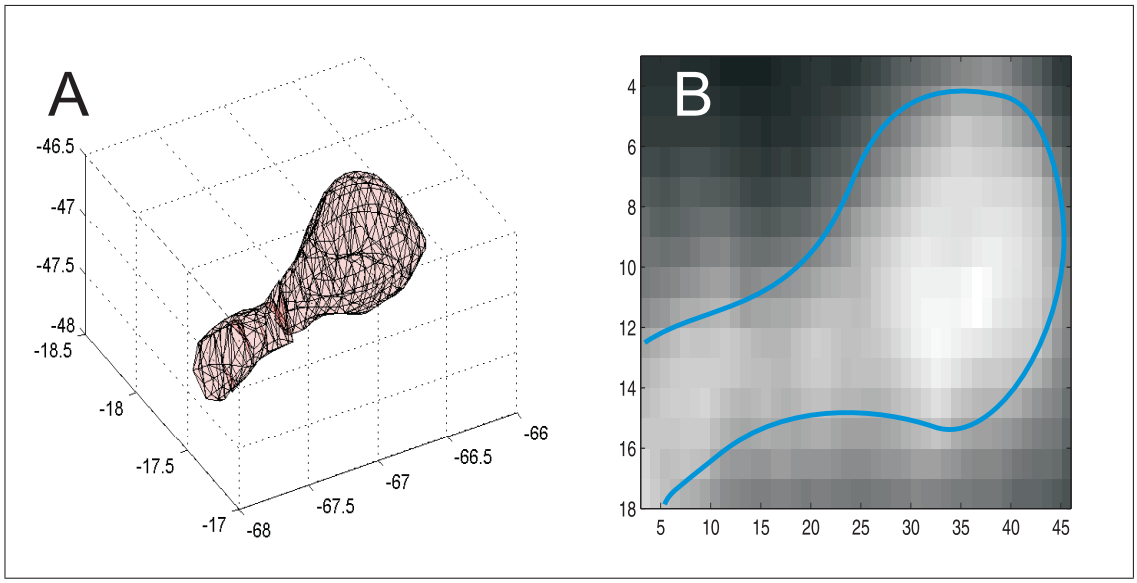

Figure 2.2: Image of a correctly segmented dendritic spine extracted from a basal dendrite. A) Dendritic spine body mesh. B) Region of interest showing the contour of the dendritic spine. Pixel size = 37.5 nm x 37.5 nm

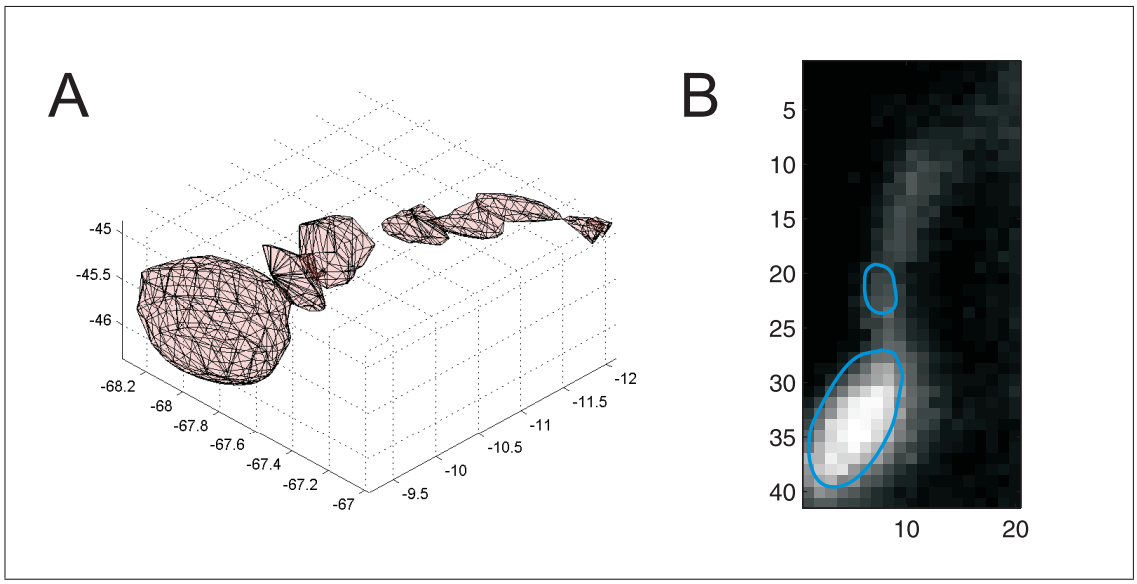

Figure 2.3: Image of a badly segmented dendritic spine extracted from a basal dendrite. A) Dendritic spine body mesh. The reader can appreciate several pieces of the broken spine. B) Region of interest showing the contour of the broken dendritic spine. Pixel size = 75 nm x 75 nm

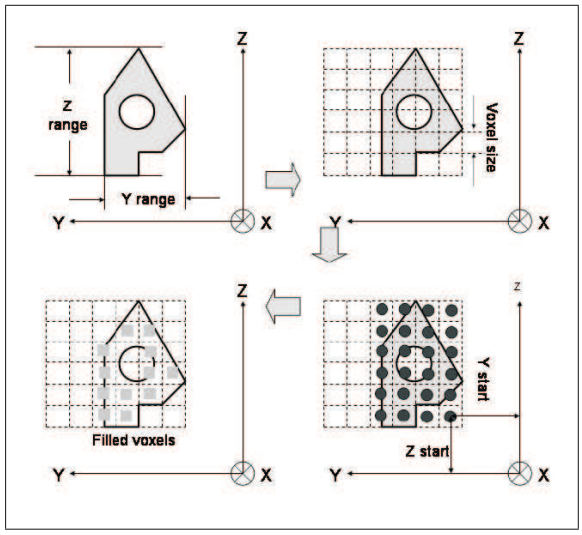

Figure 2.4: Obtained from Patil and Ravi (2005). Voxelisation by ray intersection with model.

2.3 3D Spine Reconstruction

Introduction

Our main objective was to repair the broken spines trying to preserve the semiautomatic segmentation made by the researchers of the Cajal's Institute. The first step to process the Spines is to transform them from the mesh format to a set of voxels. This procedure of converting geometric objects from their continuous geometric representation to a set of voxels is called voxelisation. Meshes of spines are composed by triangular facets representing the object surface where each facet stores the coordinate value of three vertices and a separate structure stores the connectivity between them forming the edges. Patil and Ravi (2005) proposed a voxalisation algorithm that essentially passes rays along the X-axis and finds their intersections with the facets. Voxels lying between an odd and the next even intersection are considered to be filled. This process is carried out layer-by-layer from the minimum to the maximum Z coordinate of the model (see Figure 2.4).

For the second step, we have developed a curve evolution algorithm for 3D structures. Basically, the algorithm will evolve the voxelized broken Spine following the intensity of the stack of images until a single volume is reached. This algorithm is an improvement of the 2D morphological snake algorithm presented by our group in Álvarez et al. (2010).

Active Contours and Level Sets

Snakes and their geometrically sound alternative geodesic active contours are possibly the most popular curve evolution algorithms. By iteratively solving a partial differential equation (PDE), the curve or snake deforms its shape so as to minimize internal and external energies along its boundary. The internal component keeps the curve smooth, while the external component attaches the curve to image structures, such as edges, lines, etc. This behavior makes curve evolution one of the most widely used algorithms for image segmentation and object tracking.

Let $\mathcal{C} : [0,1] \rightarrow \mathbb{R}^2$ be a parametrized 2D curve and $I : \mathbb{R}^2 \rightarrow \mathbb{R}$ be an image. In the classical active contours approach by Kass et al. (1988), the energy of the curve for an image was given by the following energy functional:

$$E(\mathcal{C}) = \int_0^1 \alpha|\mathcal{C}_p(p)|^2 + \beta|\mathcal{C}_{pp}(p)|^2 - \lambda|\nabla I(\mathcal{C}(p))|dp, \tag{2.1}$$

where $\mathcal{C}_p$ and $\mathcal{C}_{pp}$ are, respectively, the first and the second derivatives of $\mathcal{C}$ with respect to p, and α, β and λ are three positive model parameters.

The first two terms in (2.1) are the internal energy of the curve,

$$E_{int}(\mathcal{C}) = \alpha \int_0^1 |\mathcal{C}_p(p)|^2 dp + \beta \int_0^1 |\mathcal{C}_{pp}(p)|^2 dp. \tag{2.2}$$

E_{int}, which depends on the length of the curve (first term) and on its rigidity (second term), so that the smoother the curve, the lower E_{int} is.

The last term in (2.1) is the external energy, i.e., the energy of the curve not due to itself, but to the image information. The expression $|\nabla I(\mathcal{C}(p))|$ is high for points on image edges; subsequently, E_{ext} remains low on edges and gets larger in flat regions.

The snake model aims to find the curve which minimizes the functional (2.1). Such curve will be smooth and will follow some boundaries of the image. The trade-off between the smoothness and the attraction of the edges is controlled by the free variables α, β, λ. This method has been extensively applied in computer vision, especially as a good interactive tool to deal with segmentation problems. However, it suffers from several drawbacks. It heavily depends on the parameterization of the curve and relies on the selection of the three parameters. Moreover, it only looks for edges in the image and not for other structures such as lines.

The geodesic active contours (Caselles et al. (1997)) try to fix some of the problems associated with the snakes. The energy functional of the geodesic active

contours is:

$$\begin{aligned} E(\mathcal{C}) &= \int_0^{L(\mathcal{C})} g(I)(\mathcal{C}(s))ds \\ &= \int_0^1 \underbrace{g(I)(\mathcal{C}(p))}_{\text{attraction}} \cdot \underbrace{|\mathcal{C}_p|}_{\text{smoothing}} \; dp, \end{aligned} \tag{2.3}$$

where $ds = |\mathcal{C}_p|dp$ is the Euclidean arc-length parameterization of the curve, which leads to a geometric model; and $g(I) : \mathbb{R}^2 \to \mathbb{R}^+,\ \mathbf{x} \to g(I)(\mathbf{x})$ allows us to select which regions of the image we are interested in. Typically, $g(I)$ could be

$$g(I) = \frac{1}{\sqrt{1 + \alpha|\nabla G_\sigma * I|}}, \tag{2.4}$$

which is low in the edges of the image, or

$$g(I) = |G_\sigma * I| \tag{2.5}$$

which attains its minima in the centre of the image dark lines. Equation (2.3) minimises the weighted length of the curve, i.e., the optimal curve in terms of equation (2.3) has either low values of the function $g(I)$ or low values of the smoothing term along the points of its trajectory. This means the curve tends to be close to the objective areas in the image and, moreover, it is smooth and robust to undesirable effects like noise.

Minimizing the functional (2.3) consists on searching the steady state solution to the following equation:

$$\mathcal{C}_t = (g(I) \cdot \mathcal{K} - \nabla g(I) \cdot \mathcal{N}) \cdot \mathcal{N}, \tag{2.6}$$

where the right hand side is the Euler-Lagrange of the energy functional, $\mathcal{K}$ is the Euclidean curvature and $\mathcal{N}$ is the inward normal to the curve. There are two *forces* contributing in this curve evolution: a smoothing force, $g(I) \cdot \mathcal{K} \cdot \mathcal{N}$, which evolves the curve reducing its curvature; and an attraction, force $(\nabla g(I) \cdot \mathcal{N}) \cdot \mathcal{N}$, which drives the curve to the interesting areas of the image. Sometimes, the attraction force is not strong enough to move the curve (because the field $\nabla g(I)$ is too small or because this field and the curve normal are perpendicular). Hence, portions of the curve usually get stuck in these non-informative areas. In order to overcome the problem, a common solution is the introduction of the so-called *balloon force*. The curve evolution with the auxiliary balloon force is

$$\mathcal{C}_t = (g(I)\mathcal{K} + g(I)\nu - \nabla g(I) \cdot \mathcal{N}) \cdot \mathcal{N}, \tag{2.7}$$

where $\nu \in \mathbb{R}$ is the balloon force parameter.

2.3.1 The Level Set Method

The main problem of using $\mathcal{C}$ as an explicit curve is that it can not deal easily with topological changes. The Osher-Sethian (Osher and Sethian (1988)) level set method represents the curve in an implicit form as the level set of an embedding function. Let $u : \mathbb{R}^+ \times \mathbb{R}^2 \rightarrow \mathbb{R}$ be an implicit representation of $\mathcal{C}$ such that $\mathcal{C}(t) = \{(x, y); u(t, (x, y)) = 0\}$. Notice that, if the curve evolution has the form $\mathcal{C}_t = \mathcal{F} \cdot \mathcal{N}$, the evolution of any function $u(x, y)$ which embeds the curve as one of its level sets is

$$\frac{\partial u}{\partial t} = \mathcal{F} \cdot |\nabla u|. \tag{2.8}$$

Knowing that the curvature $\mathcal{K}$ can be computed with the information on the function u as $\mathcal{K} = \text{div}\left(\frac{\nabla u}{|\nabla u|}\right)$, the geodesic active contours curve evolution expression (2.7) can be written in terms of a level set implementation:

$$\frac{\partial u}{\partial t} = (g(I)\mathcal{K} + g(I)\nu + \nabla g(I) \cdot \frac{\nabla u}{|\nabla u|}) \cdot |\nabla u|, \tag{2.9}$$

where the curvature $\mathcal{K}$ can be computed with the information on the function u as

$$\mathcal{K} = \text{div}\left(\frac{\nabla u}{|\nabla u|}\right). \tag{2.10}$$

Substituting $\mathcal{K}$ in (2.9) and rearranging terms, we get the well-known curve evolution for the geodesic active contours in a level set framework:

$$\frac{\partial u}{\partial t} = g(I)|\nabla u|\left(\text{div}\left(\frac{\nabla u}{|\nabla u|}\right) + \nu\right) + \nabla g(I)\nabla u. \tag{2.11}$$

The flow given by this expression has three components, two of them related to the internal energy of the curve and one of them related to the external energy. The internal energy components are the smoothing operator, which tends to smooth the curve at high curvature segments, and the balloon force, which *inflates* or *deflates* the curve in areas of little information. The external energy component is responsible for bringing the curve to the interesting regions of the image.

Differential equation (2.11) can be solved with numerical integration methods such as a finite-difference scheme. However, these techniques are sensitive to the step size, converge slowly and may diverge.

2.3.2 Morphological Evolution of Geodesic Active Contours

Inspired by the active contour PDE (2.11), Álvarez et al. (2010) proposed a new morphological evolution method that solves the PDE but avoids the problems

of speed and convergence associated to the numerical algorithms. This new evolution use a combination of binary morphological operators whose infinitesimal behavior is equivalent to the flow expressed by Equation (2.11). Therefore, the curve is given as the zero level set of a binary piecewise constant function $u : \mathbb{R}^2 \to \{0, 1\}$. We take $u(x) = 1$ for every point x inside the curve, and $u(x) = 0$ for every point x outside the curve. The morphological operators will act on u and, hence, they will implicitly evolve the curve.

Balloon Force Operator

The erosion and dilation are two well-known morphological operators. The dilation of a function is defined as $(D_h u)(\mathbf{x}) = \sup_{\mathbf{y} \in hB} u(\mathbf{x} - \mathbf{y})$, and the erosion is $(E_h u)(\mathbf{x}) = \inf_{\mathbf{y} \in hB} u(\mathbf{x} - \mathbf{y})$. In both cases, h is the radius of the operator, and B is a disk with radius 1.

Let us study the behavior of these operators in terms of continuous-scale morphology. The function $u_d : \mathbb{R}^+ \times \mathbb{R}^2 \to \mathbb{R}$ defined as $u_d(t, \mathbf{x}) = D_t u_0(\mathbf{x})$ is the solution to the following partial differential equation:

$$\frac{\partial u_d}{\partial t} = |\nabla u_d| \tag{2.12}$$

for the initial condition $u_d(0, \mathbf{x}) = u_0(\mathbf{x})$. Therefore, D_h is the infinitesimal generator of the PDE in eq. (2.12) and we can verify that

$$\lim_{h \to 0^+} \frac{D_h u - u}{h} = |\nabla u|. \tag{2.13}$$

With a similar reasoning, we can say that the function $u_e : \mathbb{R}^+ \times \mathbb{R}^2 \to \mathbb{R}$ defined as $u_e(t, \mathbf{x}) = E_t u_0(\mathbf{x})$ is the solution to the PDE

$$\frac{\partial u_e}{\partial t} = -|\nabla u_e| \tag{2.14}$$

for the initial condition $u_e(0, \mathbf{x}) = u_0(\mathbf{x})$. These results allows to solve a level set evolution PDE like those on equations (2.12) and (2.14) using the morphological operators D_h and E_h, respectively.

We will focus on the balloon type operator term of equation (2.11):

$$\frac{\partial u_{\text{ball}}}{\partial t} = g(I) \cdot \nu \cdot |\nabla u_{\text{ball}}|. \tag{2.15}$$

The factor $g(I)$ controls the strength of the balloon force in different segments of the curve: when $g(I)$ is high, the corresponding segment is located far from a target

region, and the balloon force must be strong; on the other hand, when $g(I)$ becomes lower, the curve is approaching its objective, and hence the balloon force becomes unnecessary. The effect of the $g(I)$ factor in (2.15) can be discretized with a single threshold θ: when $g(I)$ is greater than θ, the corresponding point is updated according to the balloon force, and left unchanged otherwise. Depending on the value of ν, the remaining factors ($\nu \cdot |\nabla u_{\text{balloon}}|$) lead to the dilation and the erosion PDEs given above. Given the snake evolution at iteration n, $u^n : \mathbb{R}^2 \to \{0,1\}$, the balloon force PDE (2.15) applied over u^n can be solved using the following morphological approach:

$$u^{n+1}(\mathbf{x}_i) = \begin{cases} (D_d u^n)(\mathbf{x}_i) & \text{if } g(I)(\mathbf{x}_i) > \theta \text{ and } \nu > 0 \\ (E_d u^n)(\mathbf{x}_i) & \text{if } g(I)(\mathbf{x}_i) > \theta \text{ and } \nu < 0\,, \\ u^n(\mathbf{x}_i) & \text{otherwise} \end{cases} \tag{2.16}$$

where D_d and E_d are the discrete versions of dilation and erosion.

Smoothing Morphological Operator

Let $\mathcal{B}$ the set of all line segments of length 2 centred at the origin of $\mathbb{R}^2$. The morphological continuous line operators is defined as

$$(SI_h u)(\mathbf{x}) = \sup_{B\in\mathcal{B}} \inf_{\mathbf{y}\in\mathbf{x}+hB} u(\mathbf{y}), \tag{2.17}$$

$$(IS_h u)(\mathbf{x}) = \inf_{B\in\mathcal{B}} \sup_{\mathbf{y}\in\mathbf{x}+hB} u(\mathbf{y}). \tag{2.18}$$

The *mean operator*

$$(F_h u)(\mathbf{x}) = \frac{(SI_h u)(\mathbf{x}) + (IS_h u)(\mathbf{x})}{2} \tag{2.19}$$

has some interesting properties. The so-called Catté-Dibos-Koepfler scheme (Catt and Koepfler (1995)) relates the operator F_h with the mean curvature motion in the following manner:

$$(F_h u)(\mathbf{x}) = u(\mathbf{x}) + h^2 \frac{1}{4} |\nabla u| \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right)(\mathbf{x}) + O(h^3). \tag{2.20}$$

Rearranging terms and setting a small h, we obtain the infinitesimal generator of the F_h operator:

$$\lim_{h\to 0^+} \frac{(F_{\sqrt{4h}} u)(\mathbf{x}) - u(\mathbf{x})}{h} = |\nabla u| \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right)(\mathbf{x}). \tag{2.21}$$

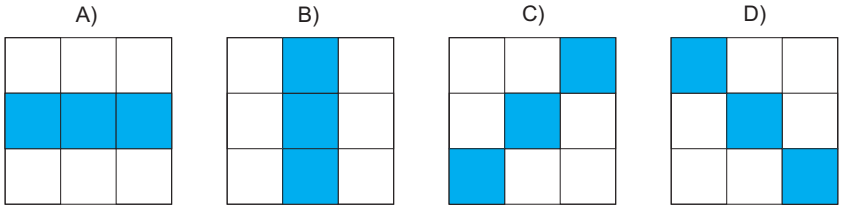

Figure 2.5: Representation of $\mathcal{P}$. See equation 2.25.

Subsequently, the mean curvature motion can be solved by means of the F_h operator. Unfortunately, one can easily see that F_h is no longer a morphological operator in the sense that it generates new level set values. We can avoid this problem using operator composition. Given any two operators T_h^1 and T_h^2, for a small h, that

$$T_{h/2}^2 \circ T_{h/2}^1 u \approx \frac{T_h^2 u + T_h^1 u}{2}. \tag{2.22}$$

Therefore, the non-morphological operator $F_{\sqrt{4h}}$ can be approximated by the morphological operator $SI_{\sqrt{h}} \circ IS_{\sqrt{h}}$. We will iterate this new operator in order to approach the solution of the weighted mean curvature PDE:

$$\frac{\partial u_{smt}}{\partial t} = g(I) \cdot |\nabla u_{smt}| \cdot \operatorname{div}\left(\frac{\nabla u_{smt}}{|\nabla u_{smt}|}\right). \tag{2.23}$$

As in the previous case, the $g(I)$ factor acts like a weight which controls the strength of the smoothing operation at every point, and we will discretize it again by means of a threshold θ. The morphological evolution of the PDE (2.23) for a known function u^n is given by

$$u^{n+1}(\mathbf{x}_i) = \begin{cases} (SI_d \circ IS_d u^n)(\mathbf{x}_i) & \text{if } g(I)(\mathbf{x}) > \theta \\ u^n(\mathbf{x}) & \text{otherwise} \end{cases}, \tag{2.24}$$

where SI_d and IS_d are the discrete versions of the above morphological continuous line operators. Both IS_d and SI_d have their own version of the set $\mathcal{B}$, $\mathcal{P}$, which is a collection of four discretized segments centered at the origin (Figure 2.5):

$$\mathcal{P} = \left\{ \begin{array}{c} \{(0,0),(1,0),(-1,0)\}, \\ \{(0,0),(0,1),(0,-1)\}, \\ \{(0.0),(1,1),(-1,-1)\}, \\ \{(0,0),(1,-1),(-1,1)\} \end{array} \right\}. \tag{2.25}$$

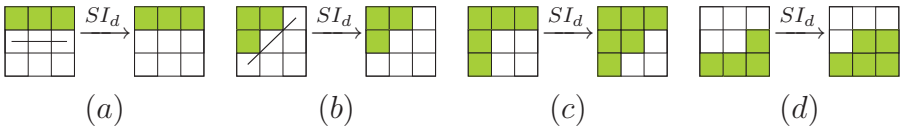

Figure 2.6: Based on Álvarez et al. (2010). Some examples of the effect of the SI_d operator on individual pixels of binary images. In those cases where a straight line is found (marked in black), the central pixel remains active ((a) and (b)). When the central pixel does not belong to a straight line of active pixels, it is made inactive ((c) and (d)). For exemplification purposes, we assume the pixels on the borders are not affected by the operator.

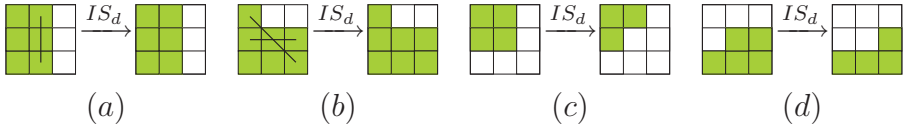

Figure 2.7: Based on Álvarez et al. (2010). Examples of the IS_d operator.

$SI_d \circ IS_d$ explanation

Álvarez et al. (2010) presented an intuitive explanation of the $SI_d \circ IS_d$ smoothing operator. In binary images u, both SI_d and IS_d perform the same operation, but SI_d works only on white (or *active*) pixels and IS_d only on black (or *inactive*) pixels. It is easy to see that SI_d does not affect inactive pixels. Suppose $u(\mathbf{x}_0)$ is an inactive pixel, ie, $u(\mathbf{x}_0) = 0$. Then, $\inf_{\mathbf{y} \in \mathbf{x}_0 + P} u(\mathbf{y})$ will be 0 for every segment P in $\mathcal{P}$, and therefore $(SI_d u)(\mathbf{x}_0) = 0$. Following a similar reasoning, we can see that IS_d does not affect active pixels.

For every active pixel $\mathbf{x}_1$ in a binary image, the SI_d operator looks for small (3 pixels long) straight lines of active pixels which contain $\mathbf{x}_1$. This search is done in the four possible orientations corresponding to the four segments in $\mathcal{P}$. If no straight line exists, the pixel is made inactive (see Figure 2.6). Sharp edges (Fig. 2.6c and 2.6d) are detected as those pixels which are not part of a straight line and removed. The active pixels in smooth edges (Fig. 2.6a and 2.6b) remain unchanged.

For inactive pixels, the IS_d operator carries out a similar procedure (see Figure 2.7). The composition $SI_d \circ IS_d$ first removes the sharp inactive pixels with IS_d, and then repeats the procedure for the active ones with SI_d. The result is a global smoothing of u, as can be seen in the Figure 2.8. Note that the examples (d) in the Figures 2.6 and 2.7 show opposite cases. Occasionally, the IS_d operator may consider an inactive pixel as *sharp*, and make it active. But then, SI_d also

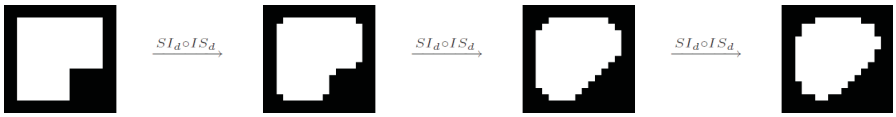

Figure 2.8: Obtained from Álvarez et al. (2010). Example of the $SI_d \circ IS_d$ operator iterated until convergence.

considers *sharp* that pixel and makes it inactive again. The composition $SI_d \circ IS_d$ leaves these pixels unchanged:

$$SI_d \circ IS_d \left(\right) = SI_d \left(\right) = . \tag{2.26}$$

Solving the complete PDE

As stated above, the active contour equation (2.11) is made up of three different components: a smoothing force, a balloon force and an attraction force. In previous sections we have seen how two of these components may be solved with morphological operators. The third component, ie, the attraction force, has an immediate discrete version as we will see shortly.

In the PDE, the combination of the three components is performed through their addition. Our morphological solution will combine them by alternating their discretized approaches: in each iteration, we will apply the morphological ballon (2.15), the morphological smoothing (2.23) and the discretized attraction force over the embedding level set function u. Given the snake evolution at iteration n, $u^n : \mathbb{R}^2 \to \{0, 1\}$, we define u^{n+1} from u^n using the following steps:

$$u^{n+\frac{1}{3}}(\mathbf{x}) = \begin{cases} (D_d u^n)(\mathbf{x}_i) & \text{if } |\nu| g(I)(\mathbf{x}_i) > \theta \quad \text{and } \nu > 0 \\ (E_d u^n)(\mathbf{x}_i) & \text{if } |\nu| g(I)(\mathbf{x}_i) > \theta \quad \text{and } \nu < 0 \\ u^n(\mathbf{x}_i) & \text{otherwise} \end{cases},$$

$$u^{n+\frac{2}{3}}(\mathbf{x}_i) = \begin{cases} 1 & \text{if } \nabla u^{n+\frac{1}{3}} \nabla g(I)(\mathbf{x}_i) > 0 \\ 0 & \text{if } \nabla u^{n+\frac{1}{3}} \nabla g(I)(\mathbf{x}_i) < 0 \\ u^{n+\frac{1}{3}} & \text{if } \nabla u^{n+\frac{1}{3}} \nabla g(I)(\mathbf{x}_i) = 0 \end{cases},$$

$$u^{n+1}(\mathbf{x}_i) = \begin{cases} (SI_d \circ IS_d u^{n+\frac{2}{3}})(\mathbf{x}_i) & \text{if } g(I)(\mathbf{x}_i) > \theta \\ u^{n+\frac{2}{3}}(\mathbf{x}_i) & \text{otherwise} \end{cases},$$

which is the morphological implementation of the active contour PDE. The new ν factor in the first step allows us to set a different threshold level for the balloon operator than for the smoothing operator. Thus we can control the strength of the balloon operator.

2.3.3 Spine Reconstruction Algorithm

The structure of the Spine Reconstruction Algorithm can be seen in Algorithm 1.

2.4 Reconstruction Validation

We have validated our results in two different ways. The first one has been the opinion of the researchers from the Cajal's Institute about the reconstruction results. For the second one, we broke several spines and applied the SRA to compare the features obtained.

2.5 Mesh Filtering

As described before, the Spine Reconstruction Algorithm grows using the pixel intensity of the stack of images, this means that the result will be a shape formed by voxels. A curve level set is calculated with an iso-surface construction algorithm in order to obtain a mesh of the spine. Due to the previous step, this surface tend to be noisy with irregular vertices that are not plausible in biology. One of the most important problems in smoothing methods is the shrinkage that happens when a shape eventually collapses to a point after iteratively applying a smoothing method. G. Taubin et al. (1995 y 1996) proposed a method for mesh smoothing based in an extension of the Fourier analysis to discrete surface signals, functions defined on the vertices of the mesh. This approach is based on a discrete approximation to the Laplacian where its eigenvectors become the "frequencies" of a given mesh. Desbrun et al. (1999) highlights the drawbacks of the previous technique: irregular connectivity meshes leads to a variety of artifacts such as geometric distortion during smoothing, numerical instability, problems of slow convergence for large meshes, and insufficient control over global behaviour. To address these problems the authors proposed an approach that contains three

Algorithm 1 Spine Reconstruction Algorithm

```
Volume and Area measure of original Spine
Voxelize
Oversampling stack Images included in the Bounding Box
if Number of unconnected components > 1 then
  Convex Hull between pair of blobs
  Intensity values incremented in ROI
  Intensity values decreased outside ROI
else
  NO SRA
end if
State = 1
while State ≠ 5 do
  if State = 1 then
    Snake grow without balloon
    if (Numberofconnectedcomponents = 1)&&(EvolvedVolume ≥
    OriginalVolume) then
      State = 2
    else if 5Iterations then
      Increaseα
    end if
  else if State = 2 then
    Smooth snake
    if (NumberBlobs ≥ 1)‖(VolumeEvol ≤ OriginalVolume) then
      State = 3
      Unevolve Spine
    end if
  else if State = 3 then
    Measure Curvature Kmax
    State = 4
  else if State = 4 then
    if Kmax ≥ 0.03 then
      α = 1.1
      Grow Snake
      MeasureKmaxEvol
      if (KmaxEvol ≥ Kmax)&&(VolumeEvol ≥ 110%Volume) then
        State = 4
      else
        Unevolve Spine
        State = 5
      end if
    end if
  else if State = 5 then
    END
  end if
  if Iterations = 1000 then
    SRA did not converge
  end if
end while
```

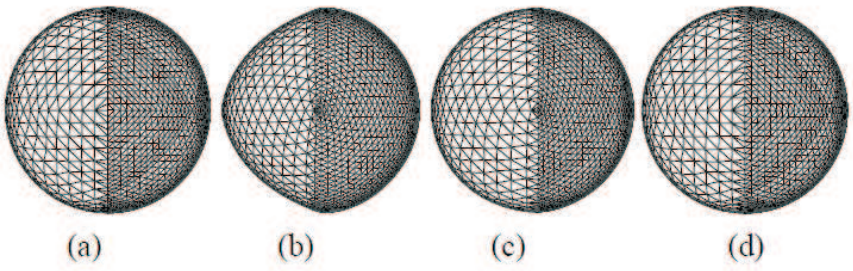

Figure 2.9: Obtained from Desbrun et al. (1999). Smoothing of spheres: (a) The original mesh containing two different discretization rates. (b) Smoothing with the Taubin (1995) method introduces sliding of the mesh and unnatural deformation, which is largely attenuated when (c) the scale-dependent version is used, while (d) curvature flow maintains the sphere exactly.

features: an implicit integration method to achieve efficiency, stability, large time-steps and a curvature flow operator that achieves a smoothing of the shape itself. A very important additional feature of this algorithm is that it includes exact volume preservation. Figure 2.9 shows the different filtering effects.

2.6 3D Global Shape Descriptors

The Imaris software (Bitplane AG) automatically provides accurate measure of the volume and the area of the Spines. Researchers of the Cajal's Institute manually marked several points to measure the Spine length and coordinates of the insertion point. After applying the spine reconstruction algortihm we have re-calculated the volume and the area of all the spines in the data base.

Pose Estimation

Spines represented as polygonal meshes are given in arbitrary orientation and position in the 3D-space, $\mathbb{R}^3$. 3D global shape descriptors can be defined in such a way that invariance with respect to translation, rotation and reflection of the spine mesh model is provided. If the invariance of descriptor with respect to similarity transforms is not provided by the representation of a feature, pose estimation is necessary as a step preceding the feature extraction. The pose normalisation procedure is a transformation of a 3D spine model I into a canonical coordinate frame by translating, rotating and reflecting the original set of vertices. One well

known tool for solving this problem is the Principal Component Analysis (PCA). But since applying PCA to a set of vertices of a mesh model can produce undesired normalisation results a modification of the PCA suggested by Vranic (2004) is used in our analysis.

Principal Component Analysis

The Principal Component Analysis (PCA) is widely used in signal processing, statistics (data analysis), compression, and neural computing (Castleman (1996)). Suppose $\mathbf{x}$ is a N-by-1 random vector that is, each element x_i of $\mathbf{x}$ is a random variable. The mean vector of $\mathbf{x}$ can be estimated from a sample of L such vectors by

$$\mathbf{m}_x \approx \frac{1}{L}\sum_{l=1}^{L}\mathbf{x}_l \tag{2.27}$$

and its covariance matrix by

$$C_x = E\{(\mathbf{x}-\mathbf{m}_x)(\mathbf{x}-\mathbf{m}_x)^T\} \approx \frac{1}{L}\sum_{l=1}^{L}\mathbf{x}_l\mathbf{x}_l^t - \mathbf{m}_x\mathbf{m}_x^t \tag{2.28}$$

The covariance matrix is N by N, real and symmetric. The diagonal elements are the variances of the individual random variables, while the off-diagonal elements are their covariances.

Now let the matrix $\mathbf{A}$ define a linear transformation that generates a new vector $\mathbf{y}$ from any vector $\mathbf{x}$ by

$$\mathbf{y} = \mathbf{A}(\mathbf{x}-\mathbf{m}_x) \tag{2.29}$$

where $\mathbf{A}$ is constructed so that its rows are the eigenvectors of $\mathbf{C}_x$. For convenience, we arrange the rows in order of decreasing magnitud of the corresponding eigenvalues. We regard the new coordinate system as the PCA coordinate system, where the data vector $\mathbf{x} \in L$ from the original system is transformed into vector $\mathbf{y}$ in the PCA frame.

The transformed vector $\mathbf{y}$, is a random vector with zero mean. Its covariance matrix is related to that of $\mathbf{x}$ by

$$\mathbf{C}_y = \mathbf{A}\mathbf{C}_x\mathbf{A}^t \tag{2.30}$$

Since the rows of $\mathbf{A}$ are eigenvectors of $\mathbf{C}_x$, $\mathbf{C}_y$ is a diagonal matrix having the eigenvalues of $\mathbf{C}_x$ along its diagonal. Thus,

$$\mathbf{C}_y = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_N \end{bmatrix}. \tag{2.31}$$

and the λ_k are the eigenvalues of $\mathbf{C}_y$ as well. Because the off-diagonal elements of $\mathbf{C}_y$ are zero, the elements of $\mathbf{y}$ are uncorrelated. Thus, the linear transformation $\mathbf{A}$ removes the correlation among the variables. Furthermore, each λ_k is the variance of y_k, the kth transformed variable.
As in Vranic (2004) we find the canonical coordinate frame of the 3D spine mesh model applying the following sequence of transformations:

1. Translation of the set I moving its center of mass to the origin of the coordinate system;

2. Rotation is applied so that the largest variance of the points is along the x-axis. Then a rotation around the x-axis is carried out so that the maximal variance in the yz-plane occurs along the y-axis (PCA);

3. Reflection with respect to xy coordinate plane is performed if the sum of certain moments is negative. Reflections with respect to yz and zx planes are fixed analogously.

Modification of the PCA

It is important to notice that due that the sizes of the triangles of the spine mesh model could significantly differ and the PCA can not be applied to the set of vertices.
Paquet (2000) proposed to use centers of gravity to form the input for the PCA where each center of gravity is multiplied by the area of the corresponding triangle, applying the PCA to the obtained set of vectors. Thus, the covariance matrix C_I (2.28) is approximated by

$$C_I = \approx \frac{1}{m} \sum_{i=1}^{m} S_I (\mathbf{g}_i - \mathbf{m}_I)(\mathbf{g}_i - \mathbf{m}_I)^T \tag{2.32}$$

To secure reflection invariance, Vranic (2004) suggest to compute a rotation matrix F formed by f_x, f_y and f_z defined by

$$f_x = \frac{1}{S} \sum_{i=1}^{m} sign(x'_{A_i} + x'_{B_i} + x'_{C_i}) S_i \left(\frac{(x'_{A_i} + x'_{B_i} + x'_{C_i})}{3} \right)^2 \tag{2.33}$$

and

$$F = diag(sign(f_x), sign(f_y), sign(f_z)) \tag{2.34}$$

Features

Tangelder and Veltkamp (2007) mention that 3D shapes can be discriminated by measuring and comparing their features. Feature based methods can be divided into four categories according to the type of shape features used: (1) global features, (2) global feature distributions, (3) spatial maps, and (4) local features. Feature based methods from the first three categories represent features of a shape using a single descriptor consisting of a d-dimensional vector of values, where the dimension d is fixed for all shapes. The value of d can easily be a few hundred. The descriptor of a shape is a point in a high dimensional space, and two shapes are considered to be similar if they are close in this space.

2.6.1 Cords-based Descriptor

As defined in Paquet (2000) a cord is a vector that goes from the center of mass of an object to the center of mass of a bounded region on the surface of the object. Most often in 3-D applications that bounded region is a triangle. The tensor of inertia of the object is then calculated using: $\mathbf{I} = [I_{qr}]$, where

$$I_{qr} = \left[\frac{1}{n} \sum_{i=1}^{n} [S_i (q_i - q_{CM})(r_i - r_{CM})] \right] \tag{2.35}$$

where $q \in \{x, y, z\}$, $r \in \{x, y, z\}$ i.e. they represent one of the x, y or z coordinates, and q_i or r_i refers to one of the coordinates of the center of mass of the i^{th} triangle, n is the number of such triangles, q_{CM} or r_{CM} refers to one of the coordinates of the center of mass of the object, and Si is the area of the surface of the i^{th} triangle. A cord can be interpreted as a slowly varying normal vector that captures regional characteristics. Then, the cord-based descriptor for a 3-D object is defined as a collection of three histograms. The first histogram represents the distribution of the angles between the cords and the first reference axis. The second histogram represents the distribution of the angles between the cords and the second reference axis. The third histogram provides the distribution of the radius. The two angle-based histograms define uniquely the orientation of the cords. The histograms are normalised using the total number of cords. The angles range from zero to 2π radians. The number of bins determines the angular resolution. For the radial histograms, the smallest value is zero and the highest value corresponds to the radius of the longest cord. The number of bins determines the radial resolution. These descriptors do not depend on the scale of the object, but only on the shape of the 3-D object. The cords-based descriptor has the same size whatever the

complexity of the object, since the descriptor size is related to the number of bins in each histogram. The descriptors can represent the object at the levels of resolution by using different number of bins in each histogram. Each bin corresponds to an interval of angles. Consequently, if the number of bins is small, the level of resolution is low, whereas if there are many bins, the level of resolution is high.

2.6.2 D2 Shape Descriptor

Osada et al. (2001) presented an approach to represent the shape signature for a 3D model as a probability distribution sampled from a shape function measuring geometric properties of the 3D model. They call this generalisation of geometric histograms a shape distribution. Samples from this distribution can be computed quickly and easily, while their hypothesis is that the distribution describes the overall shape of the represented object. The authors present five shape functions that were chosen mostly for their simplicity and invariant. In particular, they are easy to compute and produce distributions that are invariant to rigid motions. They are invariant to tessellation of the 3D polygonal model, since points are selected randomly from the surface. They are insensitive to small perturbations due to noise, cracks, and insertion/removal of polygons, since sampling is area weighted. In our experiments we have used the D2 shape function, in which the distance between two random vertices are measured (Figure 2.10). Once we have computed the shape the distributions for two objects, the dissimilarity between the objects can be evaluated using the Kullback-Leibler divergence (Kullback and Leibler (1951)).

2.6.3 Moment Based Descriptor

The shape of a 3-D object can also be described using 3-D statistical moments. Based on Paquet (2000), a 3-D statistical moment M_{qrs} is defined as

$$M_{qrs} = \sum_{i=1}^{n} S_i (x_i - x_{CM})^q (y_i - y_{CM})^r (z_i - z_{CM})^s \tag{2.36}$$

where x_i , y_i and z_i are the coordinates of the centre of mass of the i^{th} triangle, n is the total number of triangles that constitute the object, x_{CM}, y_{CM} and z_{CM} are the coordinates of the center of mass of the object, and Si is the mass of the i^{th} triangle. Statistical moments are not rotation invariant. In order to solve

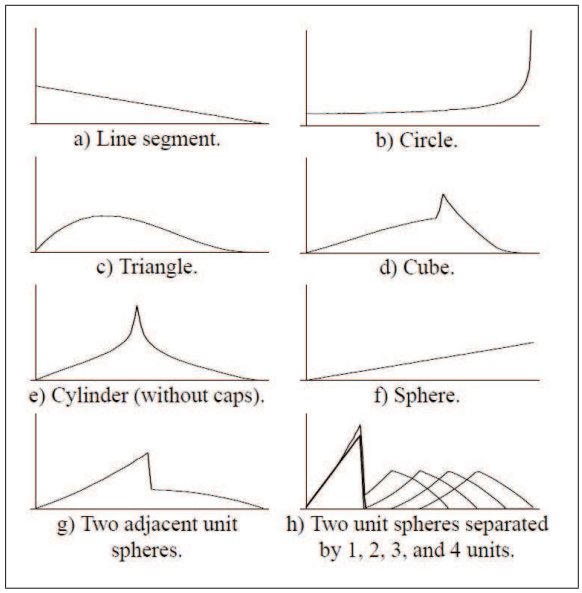

Figure 2.10: Obtained from Osada et al. (2001). Example D2 shape distributions. In each plot, the horizontal axis represents distance, and the vertical axis represents the probability of that distance between two points on the surface.

the problem they are computed in the same reference frame as the cords-based descriptors. The order of the moment is related to the level of detail. The triplet q, r, s and the value M_{qrs} uniquely define the moment. The triplet defines the order of the moment. The moments provide a statistical representation of the object. The completeness of the representation is determined by the number of moments. Low-order moments tend to describe the most important and basic characteristics while high-order moments tend to describe finer structures. Low-order moments correspond to a low-resolution representation of the object while high order moments correspond to a high-resolution representation.

Geometric moment invariants

From the moments definition, Xu and Li (2008) derives several different geometric moment invariants (GMIs). The construction process of moment invariants only needs one step by multiple integrals of invariant cores. The author proposed six 3-D moment invariant that we have implemented in order to obtain a good shape descriptor (Equations 2.37,2.38,2.39,2.40,2.41,2.42). Following the authors

notations in these equations statistical moments are named with μ_{qrs}.

$$I_1 = \frac{1}{\mu_{000}^{7/3}}(\mu_{400} + \mu_{040} + 2\mu_{220} + 2\mu_{202} + 2\mu_{022}) \tag{2.37}$$

$$\begin{aligned} I_2 \;=\; & \frac{1}{\mu_{000}^{14/3}}(\mu_{400}\mu_{040} + \mu_{400}\mu_{004} + \mu_{004}\mu_{040} \\ & +3\mu_{220}^2 + 3\mu_{202}^2 + 3\mu_{022}^2 \\ & -4\mu_{103}\mu_{301} - 4\mu_{130}\mu_{310} - 4\mu_{013}\mu_{031} \\ & +2\mu_{022}\mu_{202} + 2\mu_{022}\mu_{220} + 2\mu_{220}\mu_{202} \\ & +2\mu_{022}\mu_{400} + 2\mu_{004}\mu_{220} + 2\mu_{040}\mu_{202} \\ & -4\mu_{103}\mu_{121} - 4\mu_{130}\mu_{112} - 4\mu_{013}\mu_{211} \\ & -4\mu_{121}\mu_{301} - 4\mu_{112}\mu_{310} - 4\mu_{211}\mu_{031} \\ & +4\mu_{211}^2 + 4\mu_{112}^2 + 4\mu_{121}^2) \end{aligned} \tag{2.38}$$

$$\begin{aligned} I_3 \;=\; & \frac{1}{\mu_{000}^{14/3}}(\mu_{400}^2 + \mu_{040}^2 + \mu_{004}^2 \\ & +4\mu_{130}^2 + 4\mu_{103}^2 + 4\mu_{013}^2 + 4\mu_{031}^2 + 4\mu_{310}^2 \\ & +4\mu_{301}^2 + 6\mu_{220}^2 + 6\mu_{202}^2 \\ & +6\mu_{022}^2 + 12\mu_{112}^2 + 12\mu_{121}^2 + 12\mu_{211}^2) \end{aligned} \tag{2.39}$$

$$\begin{aligned} I_4 \;=\; & \frac{1}{\mu_{000}^{4}}(\mu_{300}^2 + \mu_{030}^2 + \mu_{003}^2 + 3\mu_{120}^2 + 3\mu_{102}^2 \\ & +3\mu_{012}^2 + 3\mu_{210}^2 + 3\mu_{021}^2 + 3\mu_{201}^2 + 6\mu_{111}^2) \end{aligned} \tag{2.40}$$

$$\begin{aligned} I_5 \;=\; & \frac{1}{\mu_{000}^{4}}(\mu_{300}^2 + \mu_{030}^2 + \mu_{003}^2 + \mu_{120}^2 + \mu_{102}^2 + \mu_{210}^2 \\ & +\mu_{021}^2 + \mu_{201}^2 + 2\mu_{021}\mu_{120} \\ & +2\mu_{300}\mu_{102} + 2\mu_{120}\mu_{102} + 2\mu_{003}\mu_{201} \\ & +2\mu_{003}\mu_{021} + 2\mu_{021}\mu_{201} + 2\mu_{030}\mu_{012} \\ & +2\mu_{030}\mu_{210} + 2\mu_{012}\mu_{210}) \end{aligned} \tag{2.41}$$

$$
\begin{aligned}
I_6 \;=\; & \frac{1}{\mu_{000}^4}[\mu_{200}(\mu_{400}+\mu_{220}+\mu_{202}) \\
& +\mu_{020}(\mu_{220}+\mu_{040}+\mu_{022}) \\
& +\mu_{002}(\mu_{202}+\mu_{022}+\mu_{004}) \\
& +2\mu_{110}(\mu_{310}+\mu_{130}+\mu_{112}) \\
& +2\mu_{101}(\mu_{301}+\mu_{121}+\mu_{103}) \\
& +2\mu_{011}(\mu_{211}+2\mu_{031}\mu_{013})]
\end{aligned} \tag{2.42}
$$

Xu and Li (2008) validate the correctness and robustness of the above moment invariants with rotated and scaled versions of a well-known model.

2.7 Clustering

Cluster analysis or clustering, is a collection of statistical methods that allow us to group cases where we measure variables or characteristics. In other words, cases that do not present similar characteristics should be grouped in different "clusters". Importantly, these clusters have to emerge because the nature of the data and without prior information [Sierra (2007)]. There are many different clustering types, one of them is the partitional cluster where we can find, just to mention a few, the *Hierarchical Clustering*, which assumes that the data can be naturally grouped in a tree-like manner; or the K-means Clustering. An important feature of the partitional clustering is that they try to minimise or maximise some criterion; taking this into account is possible to see clustering as an optimisation problem. This leads us to the two main steps of this clustering method, the measurement of the similarity or distance within different points of the data and the determination of what function has to be optimised.

One of the most known partitional clustering method is the K-means. In this method each class is characterised by a cluster centre μ_K. Given a data point the objective is to find the minimum squared Euclidean distance between him and the cluster centre. This is calculated interactively until the cluster center (centroids) do not change. Although this method is widespread used it has two main drawbacks: on the one hand, the number of clusters has to be defined before the execution of the algorithm; on the other side, the assignment depends on the initial choices of the centroids leading to the possibility of converge to local optimums.

For our purpose we have use two different clustering methods. The well-known Hierarchical Clustering and the Affinity Propagation (AP) clustering method pro-

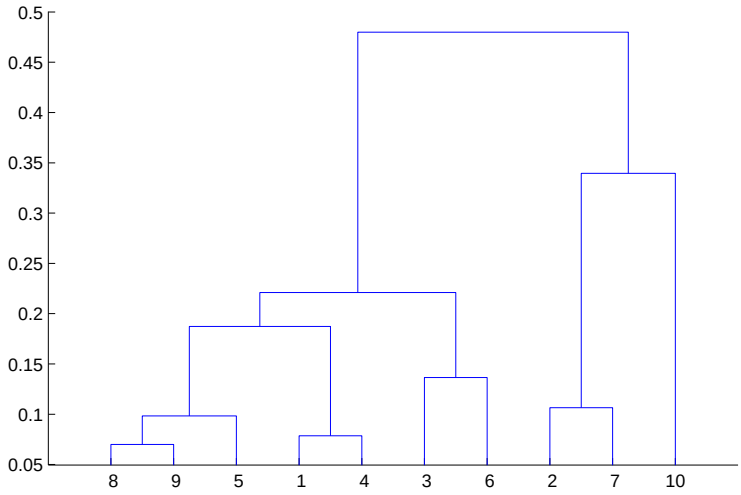

Figure 2.11: Example of a dendrogram

posed by Frey and Dueck (2007) in which real-value messages are exchanged between data points until a high-quality set of representative examples and corresponding clusters gradually emerges.

2.7.1 Hierarchical Clustering

Agglomerative clustering "bottom up", tries to build a hierarchy of a pre set number of clusters. Starting from the bottom in which every element represent a cluster and progressively a pair of cluster emerge minimising the distance between them or in other words maximising the similarity between them. This clustering method is usually represented by a dendrogram (Figure 2.11).

2.7.2 Affinity Propagation Clustering

The approach presented by Frey and Dueck (2007), tackles the drawbacks of partitional clustering methods like the K-means, the convergence to local optimums and the pre-specified number of clusters.

The affinity propagation clustering takes as input a collection of real-values similarities between data points, where the similarity $s(i, k)$ indicates how well the data point with index k is suited to be the class centre for data point i. Remarkably,

this algorithm can operate with unusual measures of similarities such as the ones that do not lie in a continuous space, the ones that are not symmetric and the ones that do not satisfy the triangle inequality. In contrast to other partitional clustering methods, AP does not need a pre-specified number of clusters. However, for each data point k a number $s(k,k)$ has to be chosen representing the prior knowledge of how good point k is a class centre. The author mention that this quantity provides a control parameter of the amount of class centres the AP is likely to find (Figure 2.13 - D). Once defined the similarity matrix S, the affinity approach can be seen as a optimisation problem, in which the search space are the valid configuration labels $c = \{c_1, c_2, \ldots, c_n\}$ and we want to minimise the energy:

$$E(c) = \sum_{i=1}^{N} s(i, c_i)$$

This optimisation is achieved by recursively transmitting messages. Two kinds of messages are exchanged between data points, "responsibility" and "availability". The responsibility $r(i,k)$, represents the accumulated evidence for how well-suited point k is to serve as the class centre for point i, taking into account other potential class centres for point i. The availability $a(i,k)$, represent the accumulated evidence for how appropriate it would be for point i to choose point k as its class centre, taking into account the support from other points that point k should be a class centre.

The AP algorithm is composed of two inputs, a main loop that iteratively updates the responsibility and the availability and the clustering output [Frey and Dueck (2007)]:

Input:

$s(i,k)$: the similarity of point i to point k. (e.i. Squared Euclidean distance)

$s(k,k)$: the preferences array which indicates the prior preference that data point is chosen as a cluster centre.

Loop:

1. Initialise the availabilities to zero:
   $a(i,k) = 0$

2. Update the responsibilities using the rule:
   $r(i,k) \leftarrow s(i,k) - max\{a(i,k') + s(i,k')\}$

3. Update the availability using the rule:
   $a(i,k) \leftarrow min\{0, r(k,k) + \sum_{i' s.t. i \notin \{i,k\}} max\{0, r(i',k)\}$

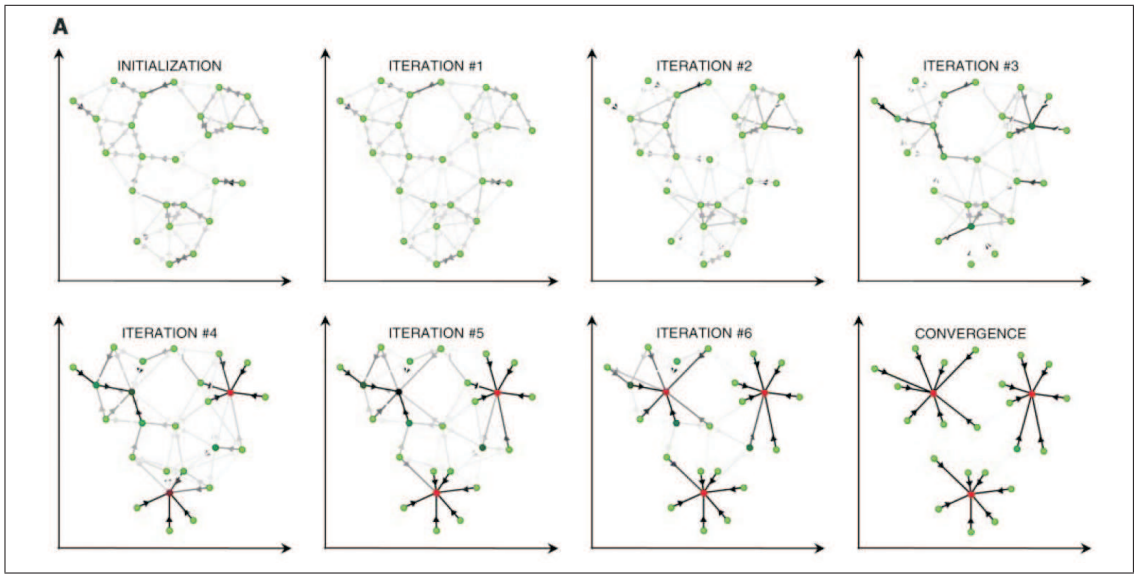

Figure 2.12: Obtained from Frey and Dueck (2007)

The self-availability is update differently:
$a(k,k) \leftarrow \sum_{i's.t.i=k} max\{0, r(i',k)\}$

4. The message-passing procedure may be finished after a fixed number of iterations, after changes in the messages fall below a threshold or after local decisions star constant for some number of iterations.

Output:

Assignments $c = \{c_1, c_2, \ldots, c_n\}$ where: $c_i = argmax_{x_k}[a(i,k)+r(i,k)]$ and c_i indexes the cluster is class centre to which point i is assigned. Specifically, if point i is in a cluster with point k serving as the class centre, then $c_i = k$ and $c_k = k$.

Figure 2.12, shows the evolution of the AP in a set of two dimensional data points. In this case, like in K-means, the squared error was used to measure similarity. We can appreciate that each point is coloured according to the current evidence that it is a cluster centre. The darkness of the arrow directed from point i to point k corresponds to the strength of the transmitted message that point i belongs to the class centre point k.

Figure 2.13, shows one step of the message-passing interaction. In b, the responsibilities $r(i,k)$ are sent from data points to candidate class centres indicating how strongly each data point favours the candidate class centre over other candidate class centres. In (c), the availabilities $a(i,k)$ are sent from candidate exemplars to data points indicating to what degree each candidate class centre is

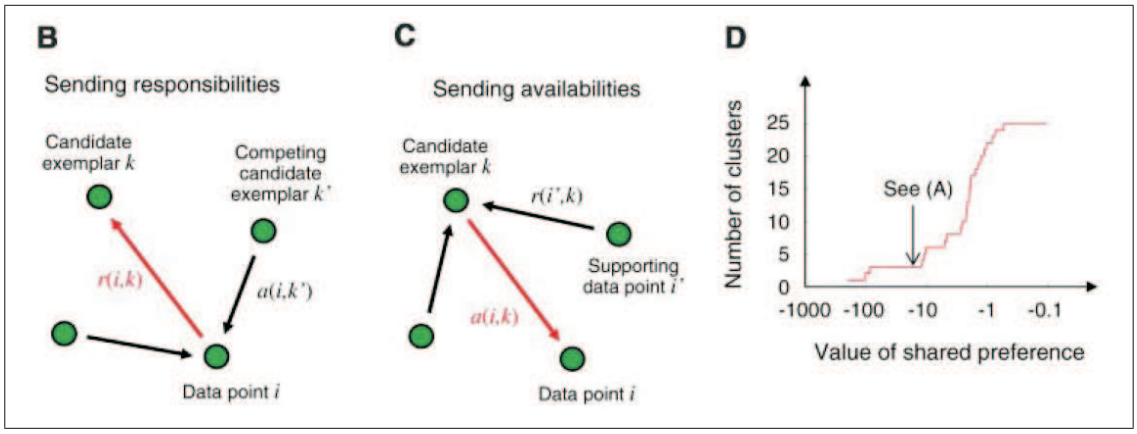

Figure 2.13: Obtained from Frey and Dueck (2007)

available as a cluster for data point. Finally, in (d) we can observe the effect of the value of the input preference on the number of identified class centres is shown.

Frey and Dueck (2007) present different comparatives within AP and other algorithms. The AP is applied to clustering images of faces (using standard squared error), clustering of putative exons to find genes(using a sparse similarity matrix), identifying key sentences in a draft (using non symmetric similarity matrix) and identifying an air-travel routing. The authors assert that the AP can find clusters with much lower error than other methods in less computational time.

2.7.3 Distance between histograms

As mentioned above, the D2 shape descriptor and the Cord-based descriptor are formed by histograms. This means that, in order to compare each spine, is necessary to define a distance between them. Kullback and Leibler (1951) defined a non-symmetric measure of the difference between two probability distributions as:

$$D_{KL}(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \tag{2.43}$$

We have used the symmetric version of this measure defined as:

$$D_{KL} = D_{KL}(P||Q) + D_{KL}(Q||P) \tag{2.44}$$

2.7.4 Clustering Assessment

There are different ways of measuring the quality assessment of the results generated by the clustering algorithms. As mentioned by Kogan (2007) there are two ways to evaluate results generated by a clustering algorithm. On the one hand the internal criteria formulate quality as a function of the given data and/or the distance-like function. On the other hand, the external criteria evaluate quality using external information such as human evaluation.

In order to compare the results obtained above we have implemented three different internal criterions: Dunn's coefficient (1974), Davies and Bauldin coefficient (1979) and Silhouette coefficient (1990).

Silhouette

This method of silhouette coefficient associates a scalar $s(a)$ with an element a of the data set $\mathcal{A}$ [Kogan (2007)]. If $\Pi = \pi_1, \ldots, \pi_k$ represents the K clusters and a $\in \pi_i$, then:

1. Compute $I(a) = \frac{1}{|\pi_i|} \sum_{x \in \pi_i} d(x, a)$, the average distance from a to other vectors in the same cluster.

2. For $j \neq i$ compute $O_j(a) = \frac{1}{|\pi_j|} \sum_{x \in \pi_j} d(x, a)$ the average distance from a to other vectors in a different cluster and let $O(a) = \min \{O_1(a), \ldots, O_{i-1}(a), O_{i+1}(a), \ldots, O_k(a)\}$ ($O_i(a)$ is omitted).

3. Compute the silhouette coefficient $s(a) = \frac{O(a) - I(a)}{\max\{O(a), I(a)\}}$.

The value of the silhouette coefficient $s(a)$ is between -1 and 1. A negative value of $s(a)$ indicates that $O(a) < I(a)$, and, perhaps, a better clustering can be achieved by placing a in a cluster other than π_i. When $s(a)$ is about 1, one has $O(a) < I(a)$ and $I(a)$ is small the cluster is said to be "dense". The silhouette coefficient for a cluster $\pi \in \Pi$ can be computed by averaging the silhouette coefficients $s(a)$ for a $\in \pi$. The silhouette coefficient for the partition Π can be computed by averaging the silhouette coefficients for all a $\in \mathcal{A}$.

Dunn's Coefficient

The coefficient proposed by Dunn (1974) is based on a partitioning $\mathcal{P}$ of the data set $\mathcal{D}$ into k clusters $C_i, i \in \{1, 2, \ldots, k\}$ a distance measure (intercluster distance) $\delta(C_i, C_j)$ between distinct clusters, and a cohesion measure (intracluster distance) $\Delta(C_i)$ for each cluster. Given this quantities Dunn's metric for the quality of a clustering is given by:

$$Q(\mathcal{P}) = \min_i \left\{ \min_{j \neq i} \left\{ \frac{\delta(C_i, C_j)}{\max_l \{\Delta(C_l)\}} \right\} \right\} = \frac{\min_i \{\min_{j \neq i} \{\delta(C_i, C_j)\}\}}{\max_l \{\Delta(C_l)\}} \tag{2.45}$$

Note that if all clusters are well-separated with respect to the intercluster distance measure $\delta(.,.)$ and tight with respect to the intracluster distance measure $\Delta(.)$, Dunn's index will assume a large positive value. Hence one approach to selecting the number of partitions k in a data set is to maximise $Q(\mathcal{P})$ with respect to k.

Davies-Bouldin Coefficient

The index of Davies and Bouldin (1979) represents a different combination of the same elements used to define Dunn's index. Specifically, given a clustering $\mathcal{P}$ and the quantities $\delta(C_i, C_j)$ and $\Delta(C_i)$ defined above, the Davies-Bouldin index is defined as

$$Q(\mathcal{P}) = \frac{1}{k} \sum_{i=1}^{k} \max_{j \neq i} \left\{ \frac{\Delta(C_i) + \Delta(C_j)}{\delta(C_i, C_j)} \right\} \tag{2.46}$$

In contrast to Dunn's index, the Davies-Bouldin index should exhibit small values for well-separated, compact clusters. Therefore, the optimum number of clusters in a data set may be determined by minimising $Q(\mathcal{P})$ with respect to k.

Kim and Billard extension

Kim and Billard (2011) suggest an extension how to measure the Dunn's and Davies-Boulin indices. They define the measure of the intercluster distance or within-cluster variance as

$$I(C_u) = \frac{1}{n_u} \sum_{i_1, i_2 = 1, i_1 < i_2}^{n_u} D^2(y_{i_1}^u, y_{i_2}^u) \tag{2.47}$$

For a partition $P_r = (C_1, \ldots, C_r)$ the authors re-define the Dunn index as

$$DI_r^S = \min_{u=1,\ldots,r} \left\{ \min_{t=1,\ldots,r, t\neq u} \left\{ \frac{I(C_t \cup C_u) - I(C_t) - I(C_u)}{\max_{l=1,\ldots,r}\{I(C_l)\}} \right\} \right\}, r = 2, \ldots, n-1 \tag{2.48}$$

and the Davis-Bouldin index as

$$DB_r^S = \frac{1}{r} \sum_{u=1}^{r} \left\{ \frac{\max_{t=1,\ldots,r, t\neq u}\{I(C_t) + I(C_u)\}}{min_{l=1,\ldots,r, l\neq u}\{I(C_l \cup C_u) - I(C_l) - I(C_u)\}} \right\}, r = 2, \ldots, n-1 \tag{2.49}$$

As the original indices the higher DI_r^S value means a better clustering outcome has occurred and a lower value of the Davis-Bouldin index implies better results for the clustering outcome.

Chapter 3

Results

3.1 Reconstruction Algorithm Validation

The original database was composed of 7759 semi-automatically segmented dendritic spines. Our algorithm has detected 1807 broken dendritic spines from those 1784 have been correctly repaired (Figure 3.1). To validate the results obtained, the researchers from the Cortical Circuits Laboratory have seen each reconstructed dendritic Spine. They have compared the shape and volume between the original spines with the reconstructed spines. Figure 3.2 shows a dendritic spine from a basal dendrite before and after applying the SRA. The original spine is broken in several pieces due to the fluorescence intensity difference between the head and the body of the spine. The SRA has been able to reconstruct the spine, obtaining a biological plausible spine shape.

3.1.1 Manually broken Spines

Although the researchers validate the reconstructed spines obtained using the SRA. We have tested our algorithm with manually broken well segmented spines. In this way, twenty complete spines has been manually broken and reconstructed using the reconstruction algorithm. Tables 3.1 and 3.2 show respectively the measurement obtained and the percentage of difference between the original spines and the evolved ones. In Figures 3.3 and 3.4, the reader can observe the results obtained after applying the algorithm in a Spine from an apical dendrite, values from this sample are highlighted in tables 3.1 and 3.2. In the same way, Figures

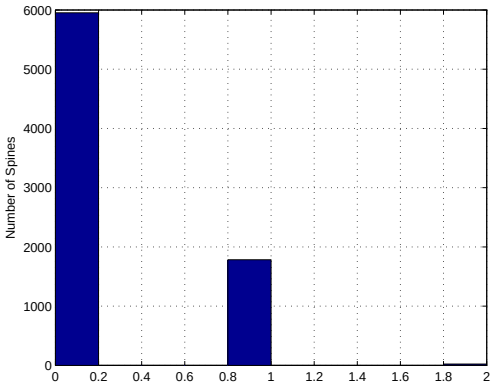

Figure 3.1: 7759 Dendritic Spines were analysed. The first column represent 5952 (76 % of the total) well segmented Spines. The second column represent the 1784 (23 % of the total)reconstructed spines using our SRA. The third column represent 23 (less than 1 % of the total) spines in which the algorithm could not reconstruct.

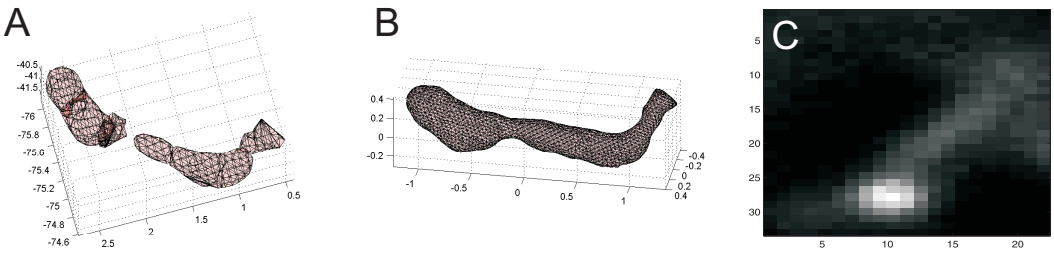

Figure 3.2: Results obtained using the Spine Reconstruction Algorithm. A) Original broken dendrite Spine from a basal dendrite. B) Reconstructed dendritic Spine after using our SRA. C) Central image of the region of interest. The reader can appreciate the intensities difference between the head and the neck of the spine.

Table 3.1: Dendritic Spines Area. Highlighted Dendritic Spines 3103 and 4699 can be observed in Figures 3.3-3.4 and Figures 3.5-3.6.

Spine	*Original Area*	*Reconstructed Area*	*broken Area*	*Diff Original - broken*	*|Diff Original - Rec|*
4640	6.10	5.87	5.49	9.99	3.69
6349	2.05	2.01	1.78	13.43	2.08
5342	4.10	4.09	3.82	6.64	0.17
5315	2.96	2.98	2.87	3.06	0.59
4699	9.44	8.90	9.39	0.53	5.75
4139	2.83	2.85	2.52	10.82	0.78
3865	5.52	5.34	4.72	14.37	3.24
3177	2.63	2.58	2.50	4.63	1.81
3103	9.07	9.18	8.28	8.63	1.24
2258	4.85	4.55	4.53	6.49	6.00
2072	7.09	7.30	6.68	5.73	3.00
1856	3.26	3.18	3.11	4.68	2.37
1516	11.16	10.79	10.21	8.44	3.27
1427	7.65	7.68	6.59	13.75	0.43
1338	4.80	5.24	3.98	16.99	9.07
264	2.72	2.67	2.35	13.31	1.61
213	6.02	5.69	5.38	10.56	5.34
5	3.16	3.30	2.81	11.00	4.54
12	2.79	2.75	2.39	14.25	1.24
72	3.42	3.44	2.97	12.89	0.74
Mean	5.08	5.02	4.62	**9.51%**	**2.85%**

3.5 and 3.6, shows the results obtained after applying the algorithm in a Spine from a basal dendrite, values from this sample are highlighted in Tables 3.1 and 3.2.

3.2 Clustering

3.2.1 Hierarchical Clustering

Hierarchical D2 Normalised + D2 Un-Normalised

As mentioned in section 2.7.3, the similarity matrix used for the clustering has been made using the symmetric Kullback Leibler divergence between the D2 shape descriptor histograms. The un-normalised D2 histogram provide information about shape and dimension of the dendritic spine, while the normalised D2 histogram just provide shape information. In order to increase the discrimination factor, a combination of both has been used to obtain the similarity matrix. In this case, the contribution of each similarity matrix is balanced and both descriptors are weighted equally.

Figure 3.7 shows the internal quality measurements for different cluster num-

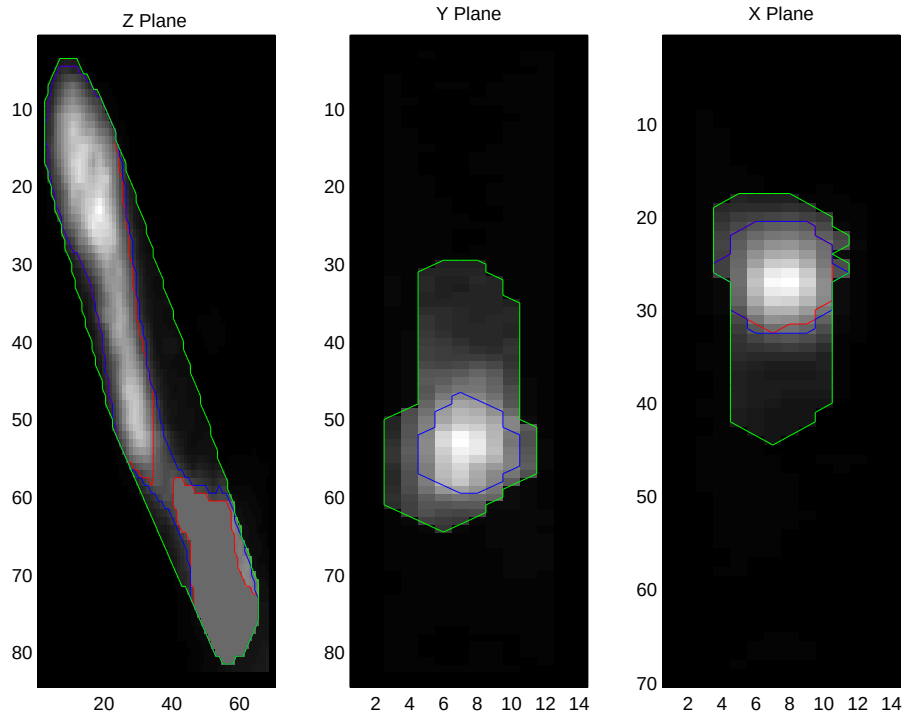

Figure 3.3: Z, Y and X Planes of the Spine 3103. The broken spine is in color red, the reconstruction in blue and the surrounding volume in green

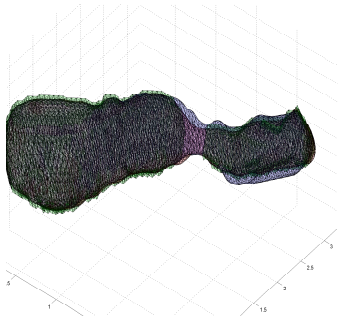

Figure 3.4: Spine 3103 Mesh. The broken spine mesh is in color green, the reconstruction in blue and the Original Spine Mesh is in pink

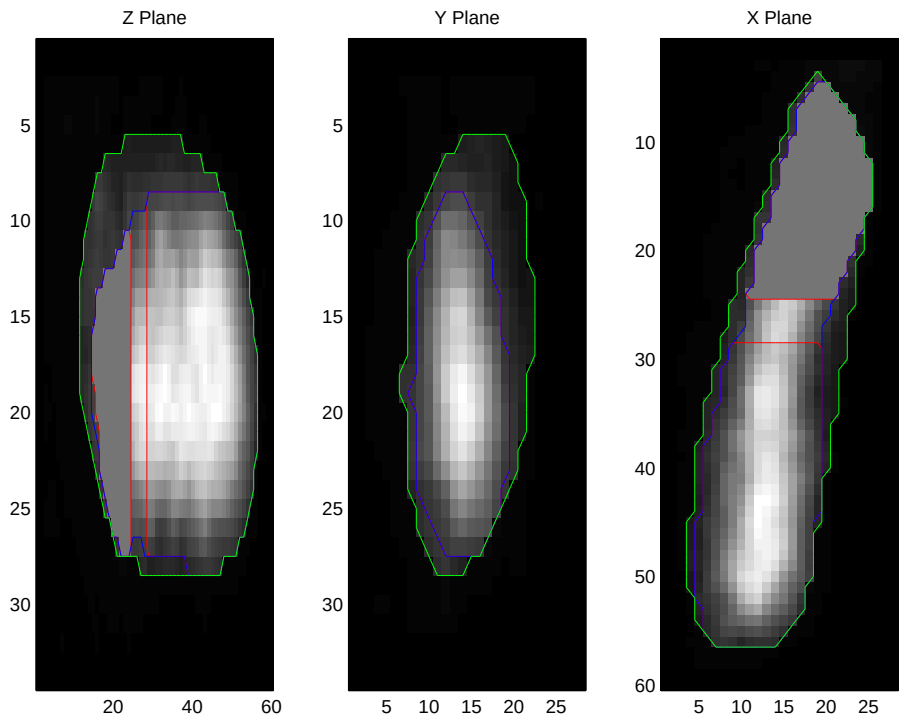

Figure 3.5: Z, Y and X Planes of Spine 4699. The broken spine is in color red, the reconstruction in blue and the surrounding volume in green

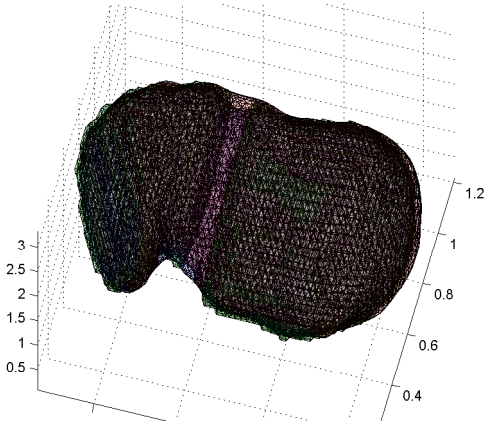

Figure 3.6: Spine 4699 Mesh. The broken spine mesh is in color green, the reconstruction in blue and the Original Spine Mesh is in pink

Table 3.2: Dendritic Spines Volume. Highlighted Dendritic Spines 3103 and 4699 can be observed in Figures 3.3-3.4 and Figures 3.5 - 3.6.

Spine	*Original Vol*	*Reconstructed Vol*	*broken Vol*	*Diff Original - broken*	*|Diff Original - Rec|*
4640	1.04	1.01	0.97	6.24	2.81
6349	0.21	0.21	0.17	17.16	0.56
5342	0.65	0.67	0.58	9.62	4.16
5315	0.34	0.36	0.27	20.52	4.78
4699	1.74	1.69	1.57	10.12	2.93
4139	0.26	0.28	0.22	15.62	8.91
3865	0.77	0.74	0.58	24.93	3.87
3177	0.30	0.28	0.25	14.42	4.52
3103	1.23	1.30	1.13	7.60	6.12
2258	0.66	0.64	0.58	12.81	3.38
2072	1.25	1.37	1.14	8.37	9.48
1856	0.36	0.39	0.29	18.34	8.02
1516	1.81	1.76	1.67	7.44	2.64
1427	1.14	1.17	1.01	10.74	3.08
1338	0.56	0.66	0.46	18.00	18.03
264	0.26	0.26	0.18	30.70	1.99
213	0.90	0.85	0.78	12.99	5.07
5	0.34	0.38	0.29	16.03	10.15
12	0.25	0.27	0.19	23.59	5.17
72	0.38	0.41	0.32	16.25	6.14
Mean	0.72	0.74	0.63	**15.07%**	**5.59%**

bers. The Davies-Bouldin index should exhibit small values for well-separated in contrast with Dunn's index that should exhibit large values. Although, a small number of cluster provides good Davies-Bouldin and Dunn's indexes, twelve clusters has been chosen. The silhuette coefficients for all the clusters have been positives, meaning that a good cluster outcome, but was not useful to discriminate between cluster numbers.

Figure 3.8 shows the amount of spines of each cluster and the Kullback-Leibler divergence between them. Figure 3.9 shows the dendritic spine that represent each cluster and Figure 3.11 shows the un-normalised D2 shape histogram.

In Figure 3.12 is possible to appreciate the clustering result in physical dimensions. The axis represent Volume, Length and Area and clusters are represented with different colours.

Finally, Table 3.3 shows the different amount of dendritic spines types that form each cluster.

Hierarchical D2 Un-Normalised + 0.5 * D2 Normalised

In this case the similarity matrix was formed using a combination of the un-normalised D2 KL-divergence matrix and a weighted normalised D2 KL-divergence matrix. This means that we have prioritised the shape and dimension (un-normalised

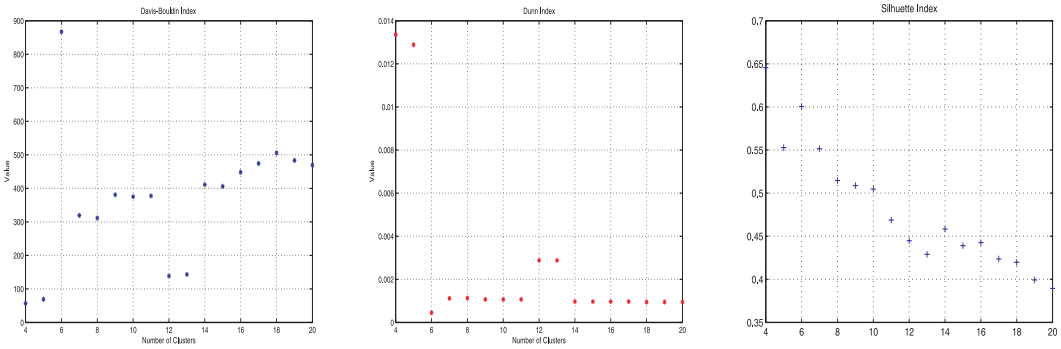

Figure 3.7: **Davis-Bouldin coefficient for clusters range from 4 to 20**. Lower value means better clustering outcome. **Dunn coefficient for clusters range from 4 to 20**. Higher values means better clustering outcome. **Silhuette index for clusters range from 4 to 20**. Values near 1 means better clustering outcome.

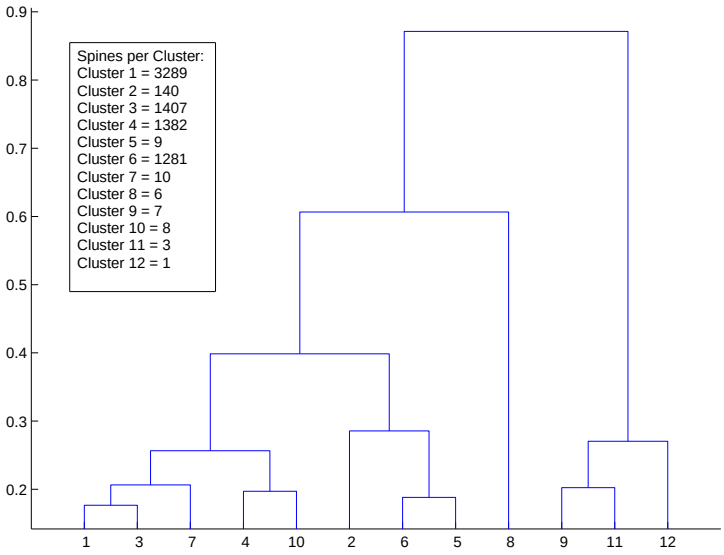

Figure 3.8: Dendogram showing connections and distance between clusters

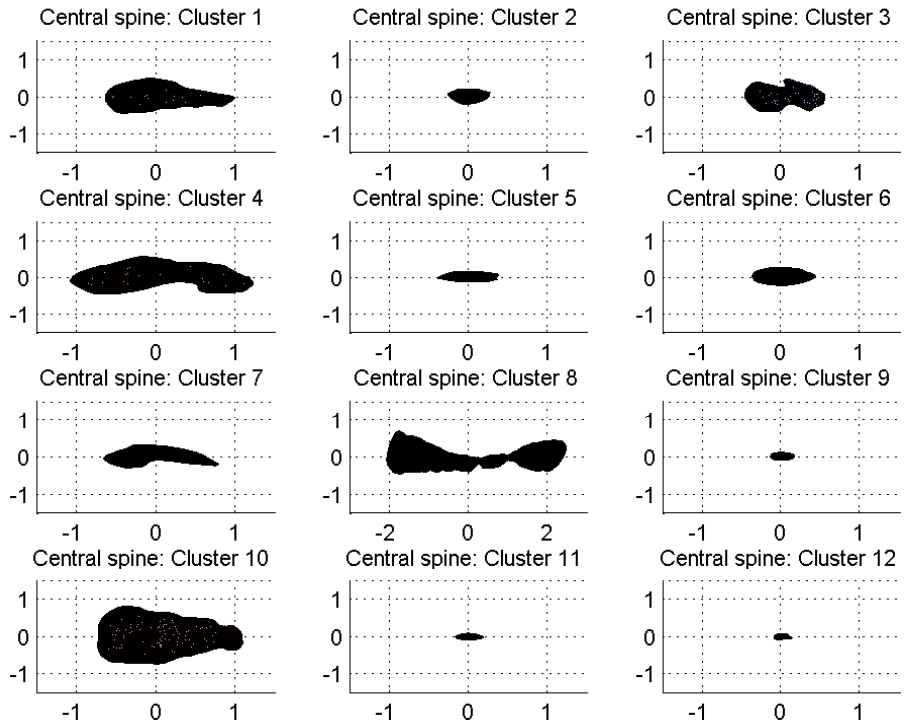

Figure 3.9: Exemplar Spines of each cluster

Table 3.3: Hierarchical clustering results with D2 and D2 normalised

				if6			m16		
Cluster Number	Spines per Cluster	Basal	Apical	Total	Basal	Apical	Total	Basal	Apical
1	3289	1863	1426	1301	817	484	1988	1046	942
2	140	110	30	45	34	11	95	76	19
3	1407	873	534	549	359	190	858	514	344
4	1382	678	704	566	288	278	816	390	426
5	9	7	2	5	4	1	4	3	1
6	1281	842	439	454	303	151	827	539	288
7	10	2	8	3	0	3	7	2	5
8	6	4	2	2	1	1	4	3	1
9	7	5	2	1	0	1	6	5	1
10	8	7	1	3	3	0	5	4	1
11	3	3	0	1	1	0	2	2	0
12	1	1	0	0	0	0	1	1	0
Sum	7543	4395	3148	2930	1810	1120	4613	2585	2028

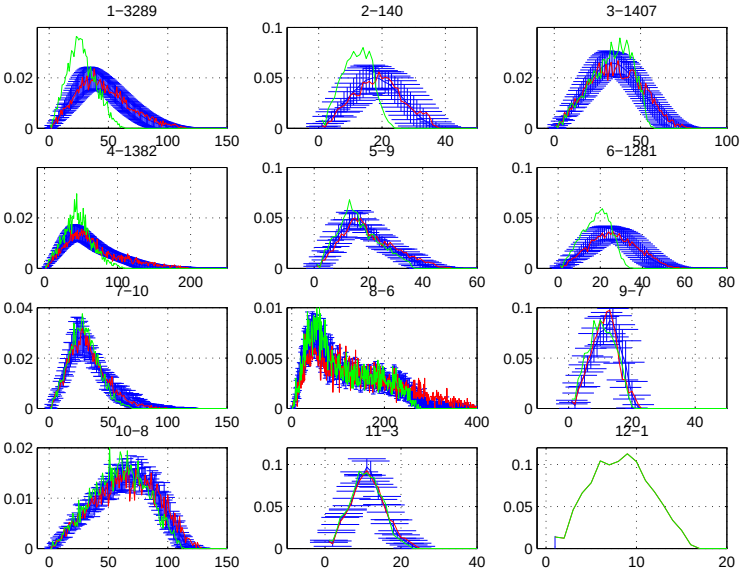

Figure 3.10: D2 descriptor in each cluster

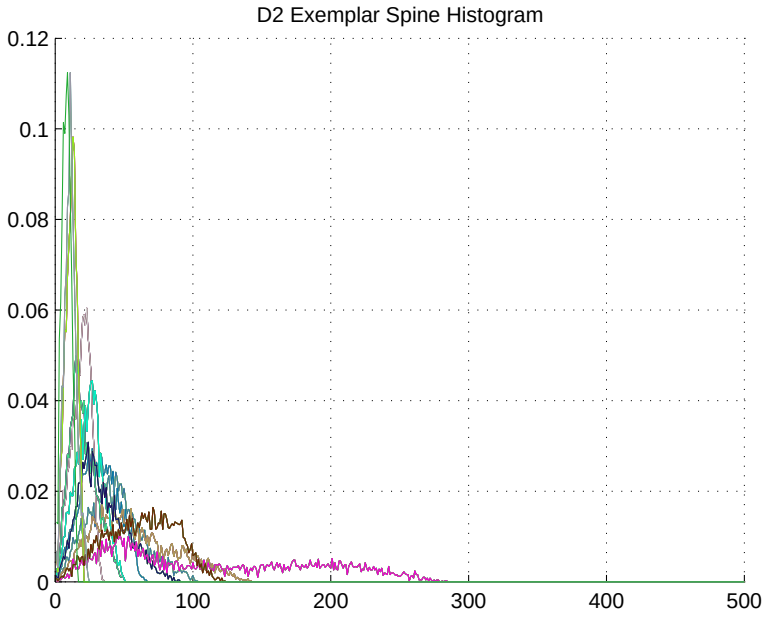

Figure 3.11: D2 descriptor for each Spine exemplar.

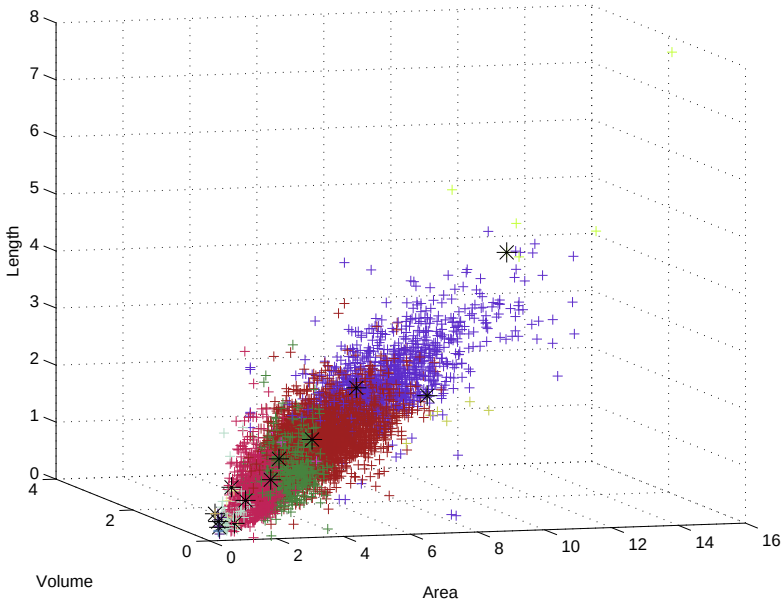

Figure 3.12: Clustering representation in the physical parameters space

D2 shape descriptor) rather just the shape (normalised D2 shape descriptor).

Figure 3.13 shows the internal quality measures, following this values we have chosen ten clusters. As in the previous clustering, the silhuette coefficients for all clusters had positive values meaning a good clustering outcome. Figure 3.14 shows the dendogram that represent the distance between clusters.

Figure 3.15 and 3.16 shows the central Spines and the D2 shape histogram respectively.

Figure 3.17 represent the clustering results in the physical dimension. This representation let us observe how the grouping has been performed.

At last, Table 3.4 represent the amount of spines in each cluster classified depending on the subject and type of dendrite.

Hierarchical un-normalized D2+Cord+Moment Invariants

In this experiment we have used a combination of different shape descriptors. The Kullback Leibler divergence similarity matrices from the D2 shape histograms and the cord histograms and the Euclidean distance between the six moment invariants. Previously to be summed, each similarity matrix has been normalised in

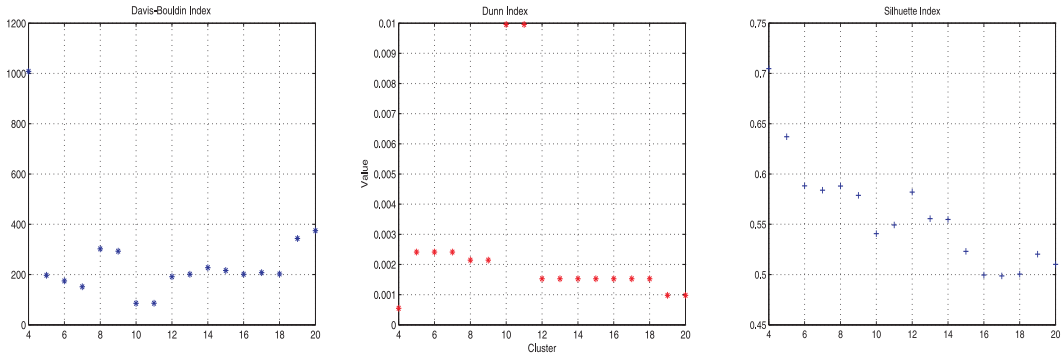

Figure 3.13: **Davis-Bouldin coefficient for clusters range from 4 to 20**. Lower value means better clustering outcome. **Dunn coefficient for clusters range from 4 to 20**. Higher values means better clustering outcome. **Silhuette coefficient for clusters range from 4 to 20**. Values near 1 means better clustering outcome.

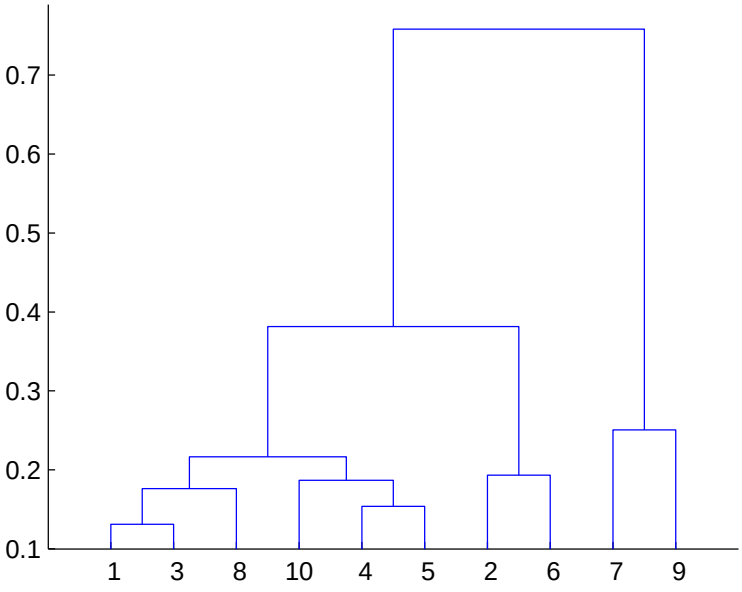

Figure 3.14: Dendogram

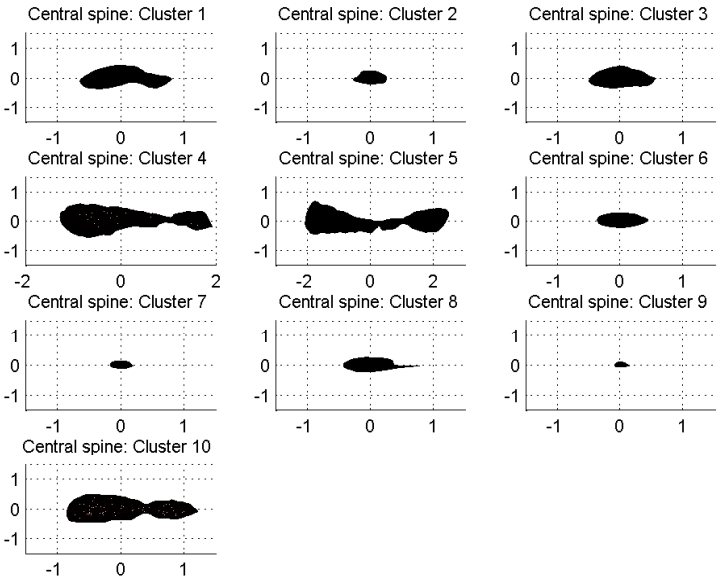

Figure 3.15: Spine exemplars of each cluster

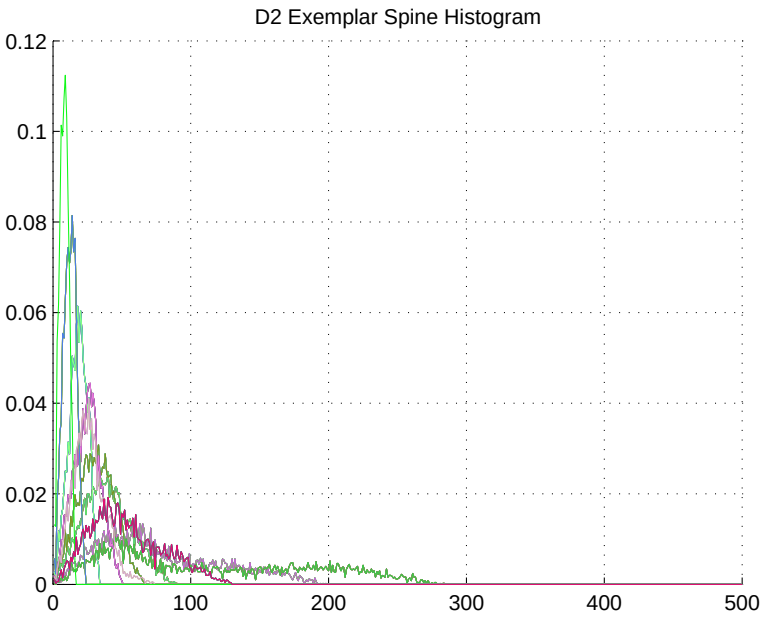

Figure 3.16: D2 descriptor for each Spine exemplar.

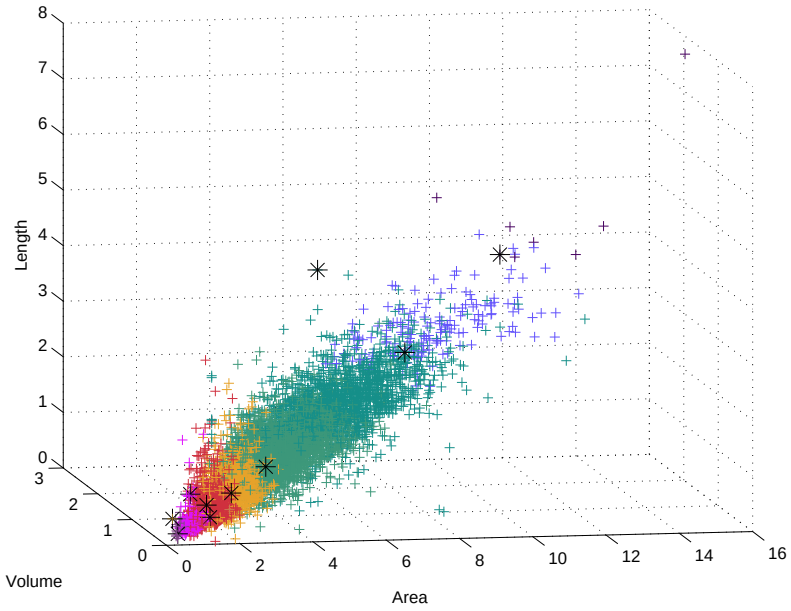

Figure 3.17: Clustering representation in the physical parameters space.

Table 3.4: Hierarchical Clustering results with D2 and 0.5 normalised D2

				if6			m16		
Cluster Number	Spines per Cluster	Basal	Apical	Total	Basal	Apical	Total	Basal	Apical
1	2706	1602	1104	1065	686	379	1641	916	725
2	151	112	39	55	37	18	96	75	21
3	1433	879	554	543	358	185	890	521	369
4	163	79	84	79	34	45	84	45	39
5	8	5	3	3	2	1	5	3	2
6	875	589	286	306	208	98	569	381	188
7	17	15	2	5	4	1	12	11	1
8	2	0	2	0	0	0	2	0	2
9	1	1	0	0	0	0	1	1	0
10	2187	1113	1074	874	481	393	1313	632	681
Sum	7543	4395	3148	2930	1810	1120	4613	2585	2028

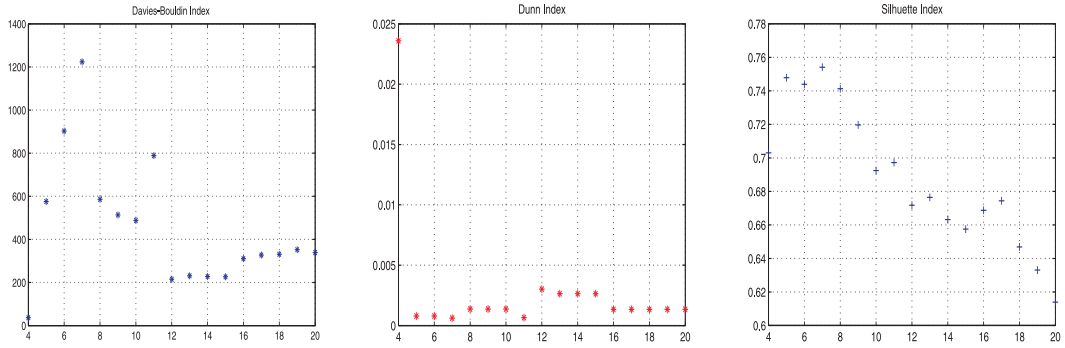

Figure 3.18: **Davis-Bouldin coefficient for clusters range from 4 to 20**. Lower value means better clustering outcome. **Dunn coefficient for clusters range from 4 to 20**. Higher values means better clustering outcome. **Silhuette coefficient for clusters range from 4 to 20**. Values near 1 means better clustering outcome.

order to have the same rage of values.

Figure 3.18 shows the internal quality measures, we have chosen twelve clusters. In Figure 3.19 the reader can appreciate the distance between these clusters.

Figure 3.20 shows the central Spines of each cluster, while the Figure 3.21 shows the clustering results in the physical dimension. As before, Table 3.5 represent the amount of spines in each cluster classified depending the subject and type of dendrite.

3.2.2 Affinity Propagation Clustering

Frey and Dueck (2007) proposed the affinity propagation cluster to find an optimal set of cluster rather than prespecifying a number of clusters or taking into account internal measures of quality. Nevertheless, it is necessary to define a value for each data point referred by the authors as "preferences". In this case, all the dendritc spines are equally suitable as exemplars so the preferences have been set to a common value. Moreover, Mézard (2007) highlights that the larger preference the more exemplars one is likely to find. In our experiments we have defined this parameter as the minimum value of the similarity matrix to obtain the smallest amount of clusters.

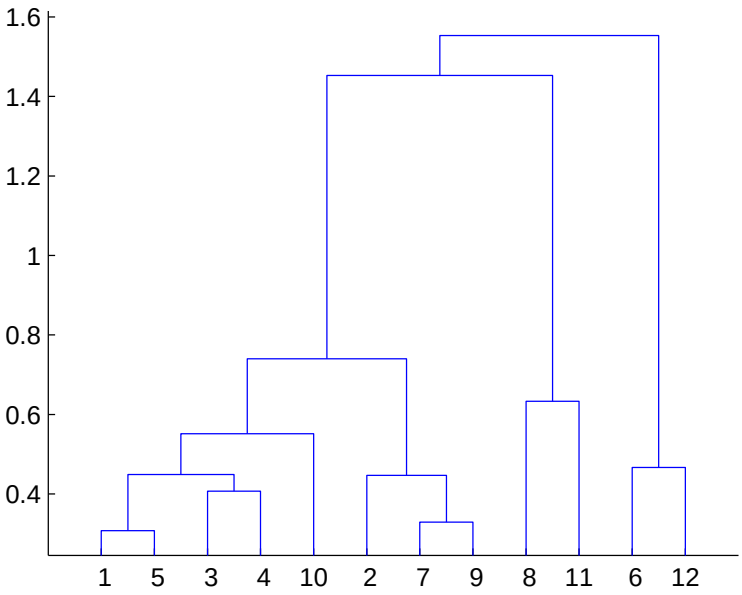

Figure 3.19: Dendogram

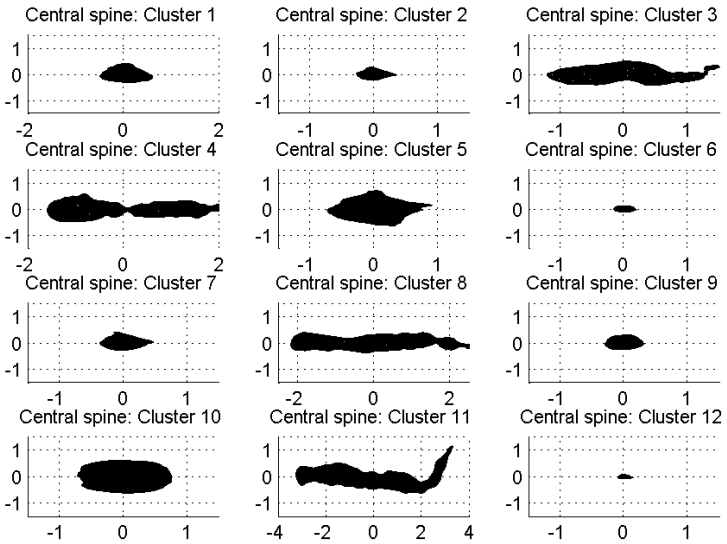

Figure 3.20: Spine exemplars of each cluster

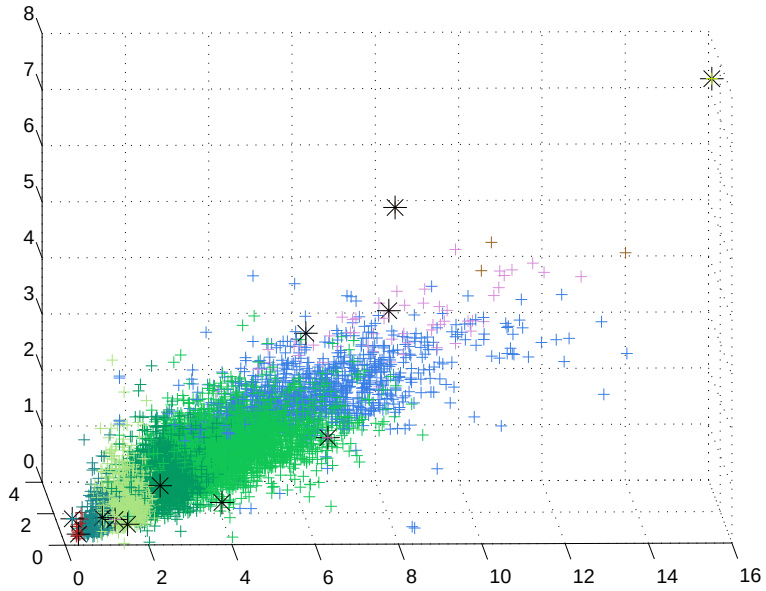

Figure 3.21: Clustering representation in the physical parameters space.

Table 3.5: Hierarchical Clustering results using un-normalized D2+Cord+Moment Invariants

				if6			m16		
Cluster Number	Spines per Cluster	Basal	Apical	Total	Basal	Apical	Total	Basal	Apical
1	1780	1056	724	701	438	263	1079	618	461
2	283	211	72	103	73	30	180	138	42
3	1105	557	548	458	242	216	647	315	332
4	55	30	25	27	13	14	28	17	11
5	3195	1799	1396	1243	771	472	1952	1028	924
6	17	15	2	5	4	1	12	11	1
7	1100	720	380	390	266	124	710	454	256
8	4	3	1	1	1	0	3	2	1
9	1	1	0	1	1	0	0	0	0
10	1	1	0	1	1	0	0	0	0
11	1	1	0	0	0	0	1	1	0
12	1	1	0	0	0	0	1	1	0
Sum	7543	4395	3148	2930	1810	1120	4613	2585	2028

Table 3.6: Clustering results using the affinity propagation with D2 + normalised D2

				if6			m16		
Cluster Number	Spines per Cluster	Basal	Apical	Total	Basal	Apical	Total	Basal	Apical
1	328	215	113	119	86	33	209	129	80
2	329	180	149	124	77	47	205	103	102
3	286	167	119	101	63	38	185	104	81
4	105	55	50	53	27	26	52	28	24
5	251	145	106	95	57	38	156	88	68
6	311	193	118	122	83	39	189	110	79
7	311	155	156	124	76	48	187	79	108
8	382	207	175	161	95	66	221	112	109
9	228	103	125	100	58	42	128	45	83
10	253	145	108	93	59	34	160	86	74
11	179	122	57	71	47	24	108	75	33
12	320	162	158	145	90	55	175	72	103
13	117	71	46	51	33	18	66	38	28
14	199	119	80	84	54	30	115	65	50
15	415	237	178	157	90	67	258	147	111
16	395	249	146	173	123	50	222	126	96
17	346	222	124	143	97	46	203	125	78
18	258	182	76	100	63	37	158	119	39
19	400	204	196	136	64	72	264	140	124
20	175	131	44	51	37	14	124	94	30
21	197	95	102	89	42	47	108	53	55
22	336	199	137	137	89	48	199	110	89
23	193	139	54	53	37	16	140	102	38
24	226	118	108	82	42	40	144	76	68
25	560	312	248	202	128	74	358	184	174
26	78	57	21	32	22	10	46	35	11
27	58	50	8	17	13	4	41	37	4
28	10	8	2	2	1	1	8	7	1
29	297	153	144	113	57	56	184	96	88
Sum	7543	4395	3148	2930	1810	1120	4613	2585	2028

Affinity D2 Normalised + D2 Unnormalized

The same similarity matrix defined for the hierarchical clustering has been used with this method to compare and analyse the results. The contribution of the normalised and the un-normalised D2 shape histograms were equally weighted to obtain the similarity matrix used in the affinity propagation clustering.

Table 3.6 shows the twenty-nine different clusters obtained by the AP and the classification depending of the sample type.

Figures 3.22 and 3.23 show the shape and D2 shape histogram of the cluster exemplars. Is easy to notice the difference in shape and size of the dendritic spines represented in the same visual scale.

Figure 3.25 represents the clustering results obtained in the physical parameters dimensions. Each color represent a cluster and the axis are volume, area and length of the dendritic spines.

Finally, figure 3.24 is a dendogram representing the distance between the ex-

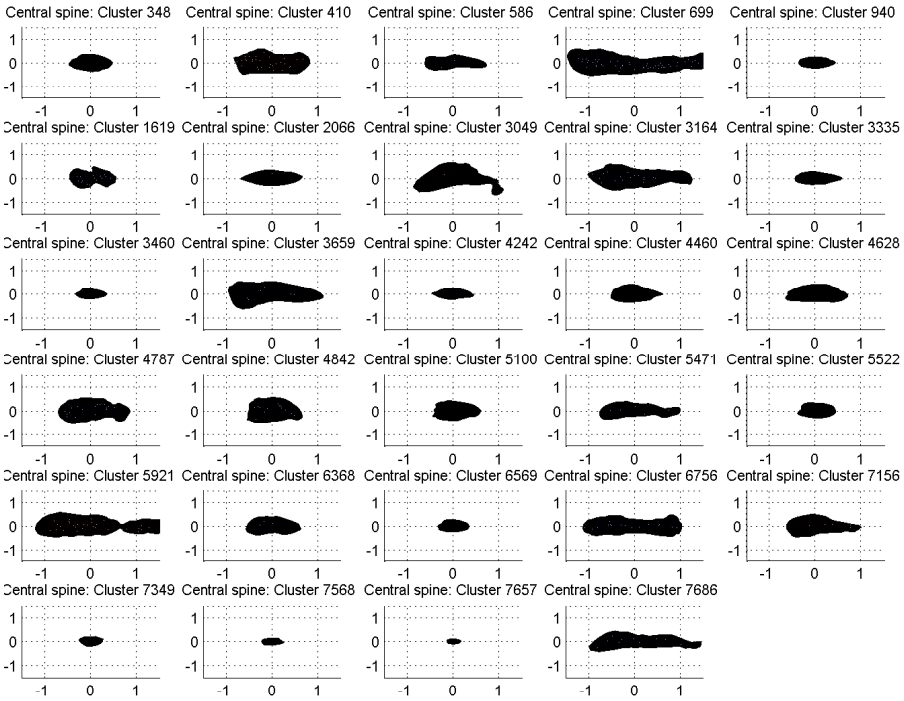

Figure 3.22: Exemplars

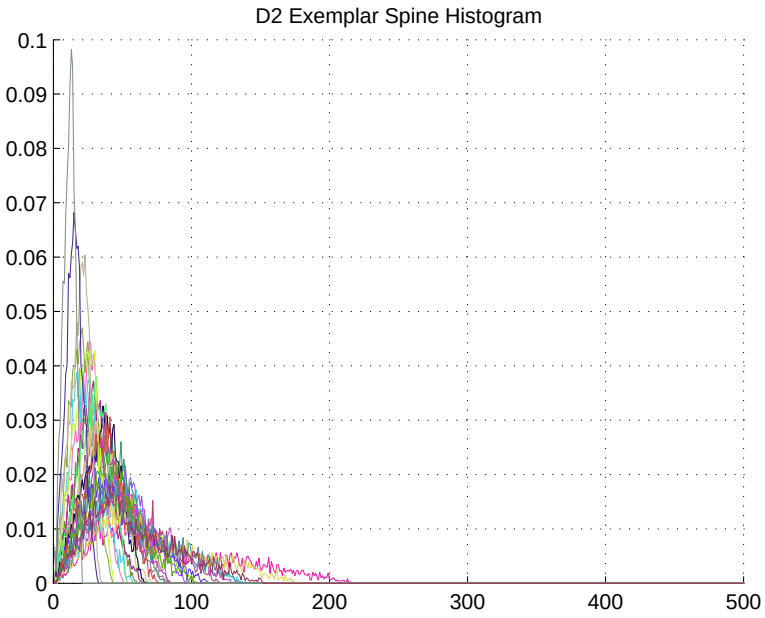

Figure 3.23: D2 shape descriptor of the exemplars

emplars. With this information the experts could decrease the amount of clusters by grouping low distance clusters.

Affinity D2 Un-Normalised + 0.5 * D2 normalized

In this experiment, the contribution of the normalised D2 shape histograms has been weighted by 0.5 to increase the information provided by the un-normalised D2 shape histograms. This similarity matrix was used in the affinity propagation clustering, which converge to twenty-six clusters. The amount of dendritic spines and the classification can be seen in Table 3.6.

The exemplars dendritic spines can be observed in figure 3.26 while Figure 3.28 shows each D2 un normalised histogram. As in the previous case, Figure 3.27 represent the distance between different exemplars.

Finally, Figure 3.29 represent the clustering results in the physical parameters dimension.

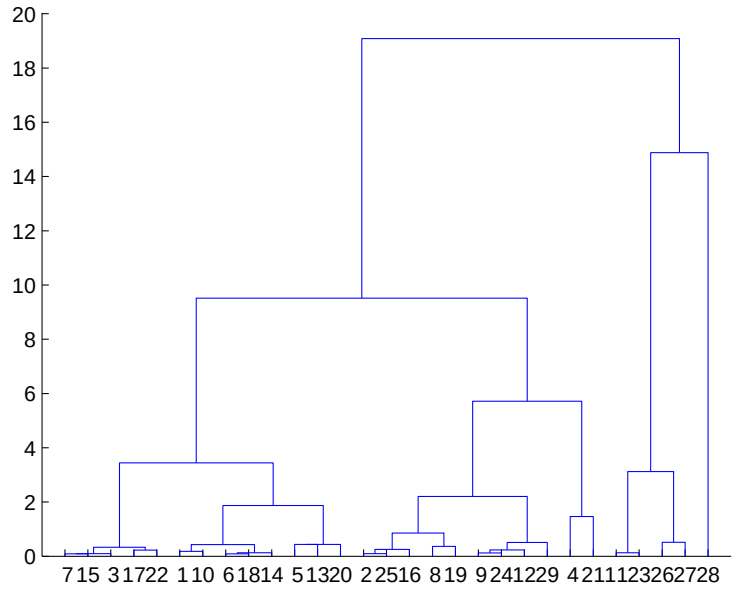

Figure 3.24: Dendogram. After performing the affinity clustering we can apply a hierarchical clustering

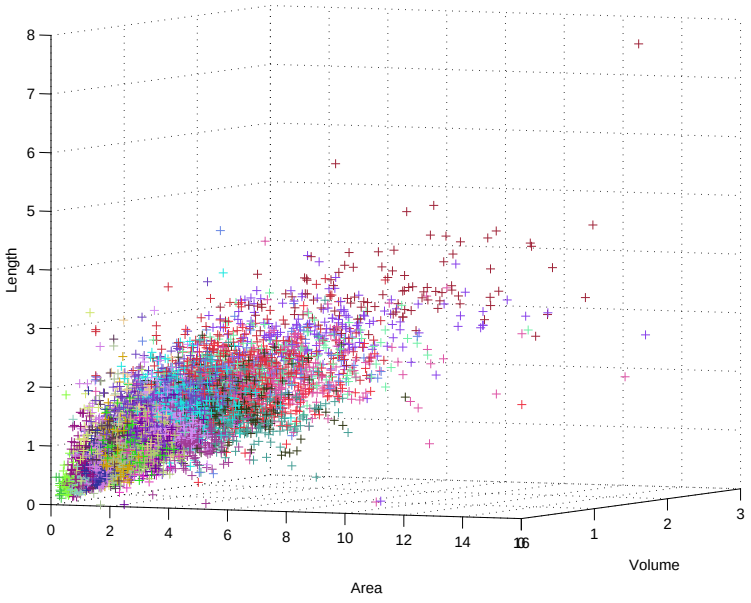

Figure 3.25: Cluster representation in physical parameters

Table 3.7: Clustering results using the affinity propagation with D2 + 0.5 *normalised D2

				if6			m16		
Cluster Number	Spines per Cluster	Basal	Apical	Total	Basal	Apical	Total	Basal	Apical
1	477	302	175	176	120	56	301	182	119
2	394	223	171	148	98	50	246	125	121
3	102	54	48	51	27	24	51	27	24
4	254	157	97	105	69	36	149	88	61
5	407	236	171	155	98	57	252	138	114
6	296	149	147	113	68	45	183	81	102
7	306	144	162	98	43	55	208	101	107
8	180	121	59	58	36	22	122	85	37
9	358	178	180	148	85	63	210	93	117
10	214	131	83	90	61	29	124	70	54
11	205	146	59	71	49	22	134	97	37
12	279	148	131	134	92	42	145	56	89
13	389	232	157	137	80	57	252	152	100
14	432	265	167	184	127	57	248	138	110
15	311	205	106	134	90	44	177	115	62
16	363	250	113	145	97	48	218	153	65
17	410	210	200	142	64	78	268	146	122
18	181	86	95	82	35	47	99	51	48
19	187	86	101	100	60	40	87	26	61
20	545	320	225	227	148	79	318	172	146
21	228	162	66	66	45	21	162	117	45
22	565	304	261	203	126	77	362	178	184
23	84	59	25	36	24	12	48	35	13
24	56	49	7	15	12	3	41	37	4
25	12	10	2	2	1	1	10	9	1
26	308	168	140	110	55	55	198	113	85
Sum	7543	4395	3148	2930	1810	1120	4613	2585	2028

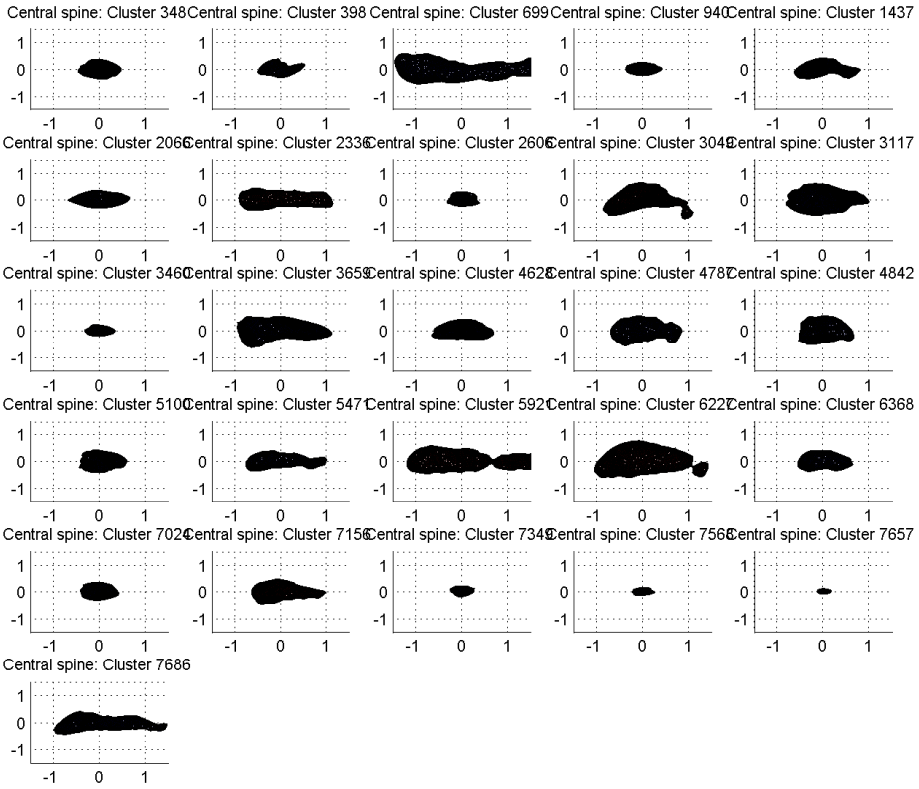

Figure 3.26: Exemplars

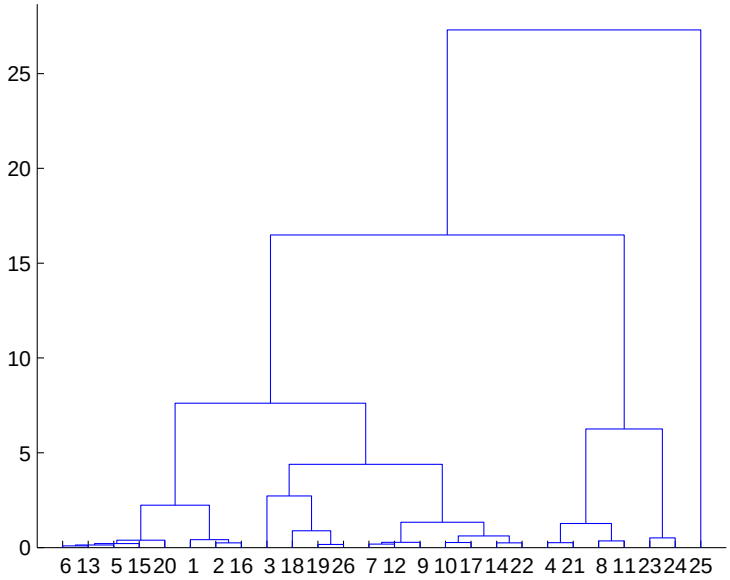

Figure 3.27: Dendogram. After performing the affinity clustering we can apply a hierarchical clustering

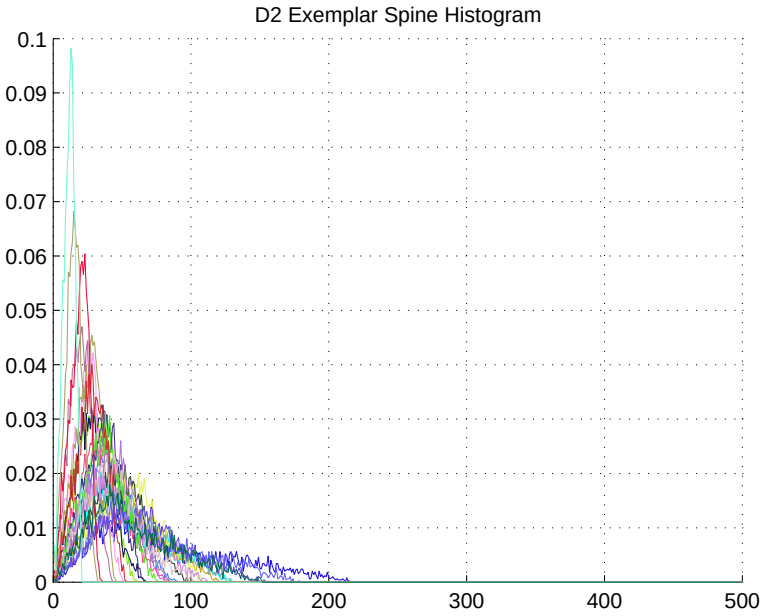

Figure 3.28: D2 shape descriptor of the exemplars

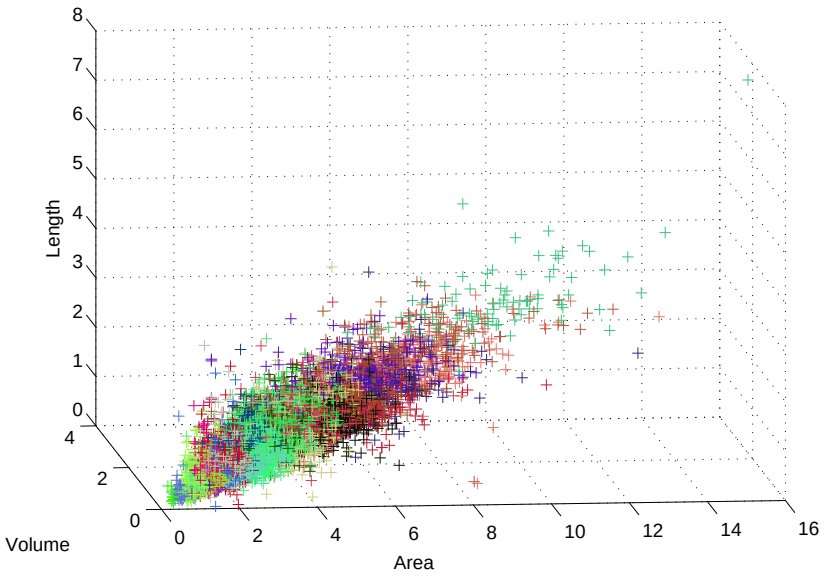

Figure 3.29: Cluster representation in physical parameters

Chapter 4

Discussion

4.1 Dendritic Spine Reconstruction

The results obtained for the 1807 reconstructed spines have been compared and analysed by the researchers of the Cortical Circuits Laboratory. They validated the results obtained and a small 6,7% (122 dendritic spines) have been marked as badly reconstructed. This classification has been done taking into account plausible biological shape, length and volume.

During experimentation, we have found that many well segmented dendritic spines present shape problems, such as tiny holes or irregular surfaces. In order to solve these problems we applied modification of our SRA, in which we just smoothed the dendritic spine (see State=2 of Algorithm 1).

4.2 Shape Descriptors

As far as we know, three-dimensional shape analysis is a quite new field. As mentioned by Tangelder and Veltkamp (2007), Xu and Li (2008), Paquet (2000), Osada et al. (2001), Vranic (2004) 3D shape retrieval methods are of mayor interest on the Internet and in domain specific databases like the automotive industry. It is possible to find many different 3D shape descriptors in literature. We have chosen the D2 shape descriptor proposed by Osada et al. (2001), the cord-based shape descriptor proposed by Paquet (2000) and the well known moment based shape descriptor with the moment invariant proposed by Xu and Li (2008). It is

important to notice that shape descriptors results must be accepted by the Biological community and descriptors like Fourier extension for 3D shapes (Zhang and Chen) or Spherical Harmonics shape descriptors (Kazhdan et al. (2003)) could be rejected due to their abstractness.

4.3 Clustering

As mentioned in Kogan (2007) the best way to evaluate clustering results is to compare them with a given standard. In our case, golden standards are not available meaning that the results must be analysed by the biological researchers.

As we have presented in the results section, three different combinations of similarity matrices have been tested. First, the sum of the Kullback Leibler divergence between the un-normalised and normalised D2 shape histograms; second, the sum of the Kullback Leibler divergence between un-normalised and a weighted normalised D2 shape histograms; and at last, the combination of the Kullback Leibler divergence between the un-normalised D2 shape histograms and the cord-based shape descriptor histogram, plus the Euclidean distance between the invariant moments.

4.3.1 Hierarchical Clustering

Due to the lack of golden standards we have implemented three different clustering assessment coefficients: Dunn's (Dunn (1974)), Davies-Bouldin (Davies and Bouldin (1979)) and the Silhouette coefficients (Kogan (2007)). It is possible to find many other coefficients in the literature as for example the Bolshakova-Azuaje family mentioned in Pearson et al. (2004). In other words, we have been able to find optimal groups of dendritic spines but these results depend on the validation made by the experts.

4.3.2 Affinity Propagation Clustering

In contrast to the hierarchical clustering, the affinity propagation finds the optimal number of clusters. However, we must set the preferences for each data point to be an exemplar. We have chosen the minimum value of the similarity matrices in order to have the least number of clusters possible. In this case,

we have found approximately thirty clusters which seem to be a large number for the biologists. With the results obtained from the affinity propagation we measured the Kullback Leibler divergence between the exemplars and performed a new hierarchical clustering. Again, these results must be analysed and verified by the biological researchers.

4.4 Biological significance

Duan et al. (2003) have demonstrated a statistically significant age-related decrease in spine number and density on both apical and basal dendritic arbors in cortical projection neurons of macaques. Fiala et al. (2002) extend these results to humans and also demonstrated that spine distributions, including spine number and distorted spine shape, are altered in the developing brain following malnutrition, alcohol or toxin exposure, infection, and in a large number of genetic disorders that result in mental retardation, such as Down's and fragile-X syndromes. As can be seen in Tables 3.3, 3.4, 3.5, 3.6 and 3.7, we have obtained the same results, in the sense that the number of dendritic spines is larger in the young (M16) subject than in the older one (IF6), both in basal and apical dendrites for the same amount of analysed samples.

Notice that the hierarchical clustering obtained less un-balanced clusters than the larger amount of balanced clusters obtained with the affinity propagation. On the one hand, in all of the hierarchical clusterings there are four main clusters containing more than 95% of the total spine account. On the other hand, the affinity propagation clustering found more small clusters. As mentioned previously, these results have been visually analysed by the biological researchers who preferred the hierarchical clustering over the affinity propagation clustering. It is highly debatable if their expertise in visually analysing spines can override the fact that we have use analytical methods to describe the spines and to perform the clustering analysis.

The utility of the morphology spine analysis can be appreciated in Tables 4.1, 4.2, 4.3 and 4.4 where the percentage of spines that form each cluster is represented. In summary, the affinity propagation clustering is composed by spines from different dendrite type whereas in the hierarchical clustering some of the clusters seem to be biased towards a certain type of dendritic spines. This is true particularly in clusters with lower amount of spines. For example in Table 4.2 cluster 7 presents almost 90% of the spines come from basal dendrites whereas in the table 4.1 cluster 7 presents almost 80% of spines from apical dendrites.

Table 4.1: Hierarchical clustering with D2 and D2 normalised. In bold we highlighted certain groups that are over the mean.

Cluster Number	if6-apicales [%]	if6-basales [%]	m16-apicales [%]	m16-basales [%]
1	14.72	24.84	28.64	31.80
2	7.86	24.29	13.57	**54.29**
3	13.50	25.52	24.45	36.53
4	20.12	20.84	30.82	28.22
5	11.11	44.44	11.11	33.33
6	11.79	23.65	22.48	**42.08**
7	30.00	0.00	**50.00**	20.00
8	16.67	16.67	16.67	**50.00**
9	14.29	0.00	14.29	**71.43**
10	0.00	37.50	12.50	**50.00**
11	0.00	33.33	0.00	**66.67**
12	0.00	0.00	0.00	**100.00**
Mean	14.85	24.00	26.89	34.27

Table 4.2: Hierarchical clustering with D2 and 0.5*D2 normalised. In bold we highlighted certain groups that are over the mean.

Cluster Number	if6-apicales [%]	if6-basales [%]	m16-apicales [%]	m16-basales [%]
1	14.01	25.35	26.79	33.85
2	11.92	24.50	13.91	**49.67**
3	12.91	24.98	25.75	36.36
4	27.61	20.86	23.93	27.61
5	12.50	25.00	25.00	37.50
6	11.20	23.77	21.49	**43.54**
7	5.88	23.53	5.88	**64.71**
8	0.00	0.00	**100.00**	0.00
9	0.00	0.00	0.00	**100.00**
10	17.97	21.99	31.14	28.90
Mean	14.85	24.00	26.89	34.27

Table 4.3: Affinity clustering with D2 and D2 normalised. In bold we highlighted certain groups that are over the mean.

Cluster Number	if6-apicales [%]	if6-basales [%]	m16-apicales [%]	m16-basales [%]
1	10.06	26.22	24.39	**39.33**
2	14.29	23.40	31.00	31.31
3	13.29	22.03	28.32	36.36
4	24.76	25.71	22.86	26.67
5	15.14	22.71	27.09	35.06
6	12.54	26.69	25.40	35.37
7	15.43	24.44	34.73	25.40
8	17.28	24.87	28.53	29.32
9	18.42	25.44	36.40	19.74
10	13.44	23.32	29.25	33.99
11	13.41	26.26	18.44	**41.90**
12	17.19	28.13	32.19	22.50
13	15.38	28.21	23.93	32.48
14	15.08	27.14	25.13	32.66
15	16.14	21.69	26.75	35.42
16	12.66	31.14	24.30	31.90
17	13.29	28.03	22.54	36.13
18	14.34	24.42	15.12	**46.12**
19	18.00	16.00	31.00	35.00
20	8.00	21.14	17.14	**53.71**
21	23.86	21.32	27.92	26.90
22	14.29	26.49	26.49	32.74
23	8.29	19.17	19.69	**52.85**
24	17.70	18.58	30.09	33.63
25	13.21	22.86	31.07	32.86
26	12.82	28.21	14.10	**44.87**
27	6.90	22.41	6.90	**63.79**
28	10.00	10.00	10.00	**70.00**
29	18.86	19.19	29.63	32.32
Mean	14.85	24.00	26.89	34.27

Table 4.4: Affinity clustering with D2 and 0.5*D2 normalised.In bold we highlighted certain groups that are over the mean.

Cluster Number	if6-apicales [%]	if6-basales [%]	m16-apicales [%]	m16-basales [%]
1	11.74	25.16	24.95	38.16
2	12.69	24.87	30.71	31.73
3	23.53	26.47	23.53	26.47
4	14.17	27.17	24.02	34.65
5	14.00	24.08	28.01	33.91
6	15.20	22.97	34.46	27.36
7	17.97	14.05	34.97	33.01
8	12.22	20.00	20.56	**47.22**
9	17.60	23.74	32.68	25.98
10	13.55	28.50	25.23	32.71
11	10.73	23.90	18.05	**47.32**
12	15.05	32.97	31.90	20.07
13	14.65	20.57	25.71	39.07
14	13.19	29.40	25.46	31.94
15	14.15	28.94	19.94	36.98
16	13.22	26.72	17.91	**42.15**
17	19.02	15.61	29.76	35.61
18	25.97	19.34	26.52	28.18
19	21.39	32.09	32.62	13.90
20	14.50	27.16	26.79	31.56
21	9.21	19.74	19.74	**51.32**
22	13.63	22.30	32.57	31.50
23	14.29	28.57	15.48	**41.67**
24	5.36	21.43	7.14	**66.07**
25	8.33	8.33	8.33	**75.00**
26	17.86	17.86	27.60	36.69
Mean	14.85	24.00	26.89	34.27

Chapter 5

Conclusions and Future Work

Conclusions:

1. We have implemented an algorithm to reconstruct broken dendritic spines. Although we have been able to correctly reconstruct the whole set of dendritic spines, it is important to notice that our SRA is highly dependent on the semi-automatically segmentation done by the biological researchers.

2. Three 3D global shape descriptors have been used to describe the spines. After many experiments we conclude that the D2 shape descriptor performed better than the cord-based and moment-based descriptors. Nevertheless, the D2 shape descriptor is rotation invariant and could lead to classification mistakes.

3. The D2 shape descriptor and the cord-based descriptor are formed by histograms. The amount of bins used determines the resolution of the descriptor. In our experiments we have used 500 bins.

4. Due to the fact that the histograms represent probability distributions we have used the Kullback Leibler divergence to measure the similarity between spines. However, Osada et al. (2001) used the Minkowski distance to classify between different shapes.

5. Two different clustering techniques have been used, the hierarchical and the affinity propagation. On the one hand the hierarchical clustering, formed clusters in an agglomerate fashion prefixing the number of clusters. This number has been chosen based on three internal quality measures. This issue is an un-resolved topic in machine learning literature. On the other

hand, the affinity propagation clustering was developed to find an optimal amount of clusters. But anyway, it has to be initialised with a preference parameter that we have chosen in order to have the less amount of clusters possible.

6. We have found ten and twelve clusters using the hierarchical clustering technique depending on the descriptors used. Whereas using the affinity propagation clustering we have found twenty six and twenty nine different clusters. As mentioned previously, clusters using hierarchical clustering tend to be un-balanced in contrast with the balanced clusters using the AP technique.

7. The results obtained have to be analysed and validated by the researchers of the Cortical Circuits Laboratory, but we can state that more spine types than the ones proposed by Peters and Kaiserman-Abramof (1970) can be found.

Future work:

1. A researcher needs about 6 hours to manually process one dendrite. Rodriguez et al. (2006) and Zhang et al. (2010) proposed different automated 3D dendritic spines detection and segmentation methods. However, a new algorithm is needed.

2. We have used 3D global shape descriptors to describe each Spine. Also, it is possible to apply local shape descriptors if we want to fully characterise the spine shape.

3. With these morphology results further analysis can be performed, such us the distribution of the different spines along the dendritic length or the frequency of each shape in a given dendrite.

4. The objective of the Cajal Blue Brain Project is to simulate the whole brain function so a complete mapping of the dendritic spine would provide the tools to create a virtual neuron which would be able to function as the real ones. This work would provide the basis to construct such maps.

5. In the future, this work will be also applied to the study of different pathological conditions, such as Alzheimer disease, and this could send some light on the morphological and functional aspects of these diseases.

Bibliography

Luis Álvarez, Luis Baumela, Pedro Henríquez, and Pablo Márquez-neila. Morphological Snakes. *IEEE International Conference on Computer Vision and Pattern Recognition*, 2010.

Jon I Arellano, Ruth Benavides-Piccione, Javier De Felipe, and Rafael Yuste. Ultrastructure of dendritic spines: correlation between synaptic and spine morphologies. *Frontiers in Neuroscience*, 1(1):131–43, 2007.

Vicent Caselles, Ron Kimmel, and Guillermo Sapiro. Geodesic Active Contours. *International Journal of Computer Vision*, 22(1):61–79, 1997.

Kenneth R. Castleman. *Digital Image Processing*. Prentice Hall Press, Upper Saddle River, NJ, USA, 1996.

Francine Catt and Georges Koepfler. A morphological scheme for mean curvature motion and applications to anisotropic diffusion and motion of level sets. *SIAM J. Numer. Anal*, 32:1895–1909, 1995.

David L. Davies and Donald W. Bouldin. A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, 1979.

Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. *Proceedings of the 26th Annual Conference on Computer graphics and Interactive Techniques - SIGGRAPH '99*, pages 317–324, 1999.

Huiling Duan, Susan L. Wearne, Anne B. Rocher, Aisha Macedo, John H. Morrison, and Patrick R. Hof. Age-related dendritic and spine changes in corticocortically projecting neurons in macaque monkeys. *Cerebral Cortex*, 2003.

J C Dunn. Well separated clusters and fuzzy partitions. *Journal on Cybernetics*, 4:95–104, 1974.

John C. Fiala, Josef Spacek, and Kristen M. Harris. Dendritic spine pathology: cause or consequence of neurological disorders? *Brain Research. Brain Research Reviews*, 39(1):29–54, 2002. ISSN 0165-0173.

Brendan J Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007.

Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *Interanational Journal of Computer Vision*, 1(4):321–331, 1988.

Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry processing*, pages 156–164, 2003.

Jaejik Kim and L. Billard. A polythetic clustering process and cluster validity indexes for histogram-valued objects. *Computational Statistics and Data Analysis*, 55(7):2250–2262, 2011.

Jacob Kogan. *Introduction to Clustering Large and High-Dimensional Data*. Cambridge University Press, 2007.

S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

Marc Mézard. Where are the exemplars ? *Science*, 315(14):949–951, 2007.

R Osada, T Funkhouser, B Chazelle, and D Dobkin. Matching 3d models with shape distributions. *Proceedings International Conference on Shape Modeling and Applications*, 21(4):154–166, 2001.

Stanley Osher and James A. Sethian. Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations. *Journal of Computational Physics*, 79:12–49, 1988.

E Paquet. Description of shape information for 2-D and 3-D objects. *Signal Processing: Image Communication*, 16(1-2):103–122, 2000.

Sandeep Patil and B. Ravi. Voxel-based representation, display and thickness analysis of intricate shapes. In *Proceedings of the Ninth International Conference on Computer Aided Design and Computer Graphics*, pages 415–422, 2005.

Ronald K. Pearson, Tom Zylkin, James S. Schwaber, and Gregory E. Gonye. Quantitative evaluation of clustering results using computatinal negative controls. In *In Proceedings of the SIAM International Conference on Data Mining*, pages 188–199. SIAM, 2004.

Alan Peters and Ita R. Kaiserman-Abramof. The small pyramidal neuron of the rat cerebral cortex. the perikaryon, dendrites and spines. *American Journal of Anatomy*, 127(4):0–321, 1970.

Dale Purves. *Neuroscience, Fourth Edition*. Sinauer Associates, Inc., 2008.

Santiago Ramón y Cajal. *Textura del sistema nervioso del hombre y de los vertebrados*. Moya, 1899.

Alfredo Rodriguez, Douglas B Ehlenberger, Patrick R Hof, and Susan L Wearne. Rayburst sampling, an algorithm for automated three-dimensional shape analysis from laser scanning microscopy images. *Nature Protocols*, 1(4):2152–61, 2006.

Basilio Sierra. *Aprendizaje Automatico: Conceptos Basicos y Avanzados*. Prentice Hall, 2007.

Johan W. H. Tangelder and Remco C. Veltkamp. A survey of content based 3D shape retrieval methods. *Multimedia Tools and Applications*, 39(3):441–471, 2007.

G. Taubin. Curve and surface smoothing without shrinkage. In *Proceedings of the Fifth International Conference on Computer Vision*, pages 852–. IEEE Computer Society, 1995.

Dejan V. Vranic. 3d model retrieval. Ph.D. Thesis, University of Leipzig, 2004.

Stefan Wilhelm. Confocal laser scanning microscopy. *Carl Zeiss MicroImaging GmbH*, 2010.

Dong Xu and Hua Li. Geometric moment invariants. *Pattern Recognition*, 41(1): 240 – 249, 2008.

R Yuste and T Bonhoeffer. Morphological changes in dendritic spines associated with long-term synaptic plasticity. *Annual Review of Neuroscience*, 24(1862): 1071–1089, 2001.

Cha Zhang and Tsuhan Chen. Efficient feature extraction for 2D/3D objects in mesh representation. *Proceedings 2001 International Conference on Image Processing*, pages 935–938.

Yong Zhang, Kun Chen, Matthew Baron, Merilee a Teylan, Yong Kim, Zhihuan Song, Paul Greengard, and Stephen T C Wong. A neurocomputational method for fully automated 3D dendritic spine detection and segmentation of medium-sized spiny neurons. *NeuroImage*, 50(4):1472–84, 2010.